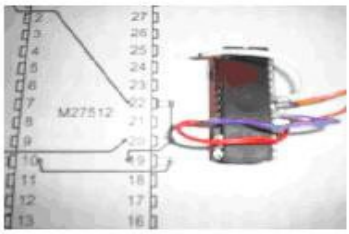
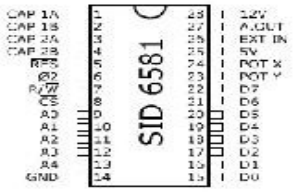
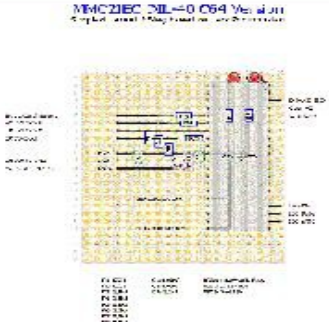
Retro-Computing

oder
25 Jahre nach C64 & Co.







Impressum

© 2008 – 2015 Peter Sieg

Author: Peter Sieg

Rabishauerstr. 9
37603 Holzminden
Germany

Stand: August 2015

Freigegeben unter Creative Commons Lizenz:

Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 de (Details)

Mit der Freigabe unter einer Creative Commons Lizenz verbindet sich der Wunsch, meine Arbeit der Community zurück zu geben in der Hoffnung, das es dort weiter gepflegt und ergänzt wird und somit zu einem aktuellen Nachschlagewerk weiter wächst.
Frei nach dem ersten Star Trek Film – V'ger :-)

Alle Informationen in diesem Buch werden ohne Rücksicht auf einen evtl. Patentschutz veröffentlicht. Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt. Bei der Zusammenstellung von Texten und Abbildungen wurde mit großer Sorgfalt vorgegangen. Trotzdem können Fehler nicht ausgeschlossen werden. Verlag, Herausgeber und Autoren können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung, noch irgendeine Haftung übernehmen.

Genannte Hard- und Softwarebezeichnungen sind ggf. auch eingetragene Warenzeichen und sollten als solche betrachtet werden. Die Rechte an verwendetem Bildmaterial liegt bei den jeweiligen Rechteinhabern.

Sollte jemand gegen Inhalte etwas einzuwenden haben, bzw. Rechte verletzt werden, so bitte ich um Nachricht. Ich werde das dann bei berechtigten Einwaenden umgehend aendern. Für alle Erweiterungen/Veraenderungen ist der jeweilige Author verantwortlich.

Inhaltsverzeichnis

Vorwort....6
Kleine C64 Reparaturanleitung....8
Benötigtes Werkzeug....8
Wenn der C64 nicht mehr funktioniert....8
Ab hier wird es komplizierter....9
C64 Netzschalter Instandsetzung/Reparatur....13
C64 Ersatz Ein-/Ausschalter....20
C64 Einbau Suppressordiode....22
C64 Bausteine (Chips)....25
Platinen Versionen....25
CBM Bezeichnungen:....27
CPU 6510....27
CIA 6526....28
VIC 6567/9....28
SID 6581....29
RAM....30
ROM....32
Eprom Pinouts....33
Messpunkte mit dem Oszi beim C64....34
C64 Ersatz-PLA auf 27512 Eprom Basis....41
Dazu folgende Erklärungen....42
Hier ein Aufbau....44
Ersatz-PLA auf CPLD Basis....45
C64 Restaurationen....46
C64 Assy 250407 Wiederbelebung....46
C64 Restauration Pentagon....50
Großumbau C64 (Verringerung der Stromaufnahme; Lebensverlängernde Maßnahmen)....58
ROM auf EPROM....60
Ersatz-PLA....62
7805 auf Schaltregler....63
7812 auf Schaltregler....65
RAM Umbau von 4164 auf 53c464-70....66
Gleichrichter auf Schottky Dioden....69
Kühlkörper nachrüsten....70
C64 – Netzteilschutzschaltung....71
Netzteil nur für C64G/-II / ASSY 250469 und VC20....73
PC Netzteil Adapter....74
S-Videoausgang am C64....77
Floppy 1541 - Wartungs- und Einstellarbeiten....78
MMC2IEC – SD2IEC: SD Card am C64....95
Historie / Hintergründe....95
Mein Nachbau....95
Tips und Hinweise....102
Neue Firmware sd2iec....104
Eagle Layout für MMC2IEC....105
IEC-ATA V2 – IDE Laufwerke am C64....108
Tapelaufwerk – Cassiopei....109
Easy Flash für C64....112
Der erste Prototyp....113
Funktion....114
DCM – Draco Cart Maker....117
NeoRAM - Batterie gepufferte RAM Disk für C64....118
Speedmessungen verschiedener Lösungen....121
1541 Ultimate....122
Ersatz für den 8701 IC....124
SwinSID – SID Ersatz auf AVR Basis....126
Micro SwinSID88....132

Tiny Eprommer Nachbau....134
T2C64 – C64 und Transputer....139
I2C Projekte....140
I2Cpeek....140
FM Radio....141
VC20 Multi Cartridge....142
Final Expansion für VC20....145
Was kann man damit machen....146
Das Konzept der LOADER-Dateien....146
Aufbau der Final Expansion v3.1....147
Neigungssensor + AVR als Joystick....151
TFT's – Kontrollmonitore für C64....158
XS1541 – AVR zwischen PC und CBM Floppy....159
XU1541 – USB Interface für CBM Floppies....164
XUM1541 – Zoomfloppy/Teensy/Open Kubus....166
Kampf dem Gilb....169
64 DTV....172
DTV Anschlußpunkte im Überblick....173
Amiga 500....180
Amiga 600....181
Amiga Custom Chips....182
Amiga Fehlercodes nach dem Einschalten....183
Amiga Akku Schaden....184
CF Bootdiskette erstellen....189
ADF auf Floppy schreiben....190
PS/2 Maus am Amiga (und Atari ST)....191
PS/2 Tastaturadapter – nicht nur für A500....192
Einfachst PIC Programmer (Ludipipo)....194
PC Floppy für Amiga umbauen....195
Unterschiede zwischen PC und Amiga Laufwerken:....195
Floppy Typen, die erfolgreich umgebaut werden können:....195
Beispiel: NEC FD1231H....197
Beispiel: MITSUMI D359T5....198
Beispiel: SAMSUNG SFD-321B....200
Floppy ‚NO Disk' Hack....202
A500+ auf 2MB Chipram Umbau....203
Amiga 500 intern auf 1MB Ram aufrüsten....204
A600 Fastramerweiterung 4/8MB....205
Floppy Emulatoren....210
Amiga RPi Floppy Emulator....212
Cortex Amiga Floppy Emulator....214
ELKO's an SMD Amiga's tauschen....221
Ersatzteilliste für Elkos im Amiga 600....222
Ersatzteilliste für Elkos im Amiga 1200....223
AT/X Netzteil für A1000....224
TICK Signal....225
Amiga am TFT anschließen....227
A520 TV-Modulator auf S-Video umbauen....229
Hier die übersetzte Schritt für Schritt Anleitung....230
Amiga 500 Tastatur am CDTV....231
Verkaufszahlen Commodore Deutschland....232
Minimig....233
Tastaturbelegung PS/2....237
Micro KBD - Micro-Tastaturersatz für Minimig....237
Sourcode....238
AVR Applikationen....242
Einfachst Programmer....242
Lochrasteraufbau....243
Magic Light....244
Tiny Basic im ATmega8 auf Pollin Eval. Board....245
AVR Nova Strike (Galaxian)....246

AVR CP/M Minirechner....248
AX81 – ZX81 im AVR....253
64'er Commander....255
64Copy....256
Tok64 – Konvertierung ASCII-Text <-> Basic Token....256
Emulationen....257
VICE....257
(Win)UAE....258
Weiterführende Adressen....259
Verein zum Erhalt klassischer Computer e.V....259
Forum64....260
C64 Wiki....260
A1k.org....261
Kleine Bauteilkunde....262
Widerstände....262
SMD Widerstände....262
ELKO'S....263
Leuchtdioden....264
Integrierte Schaltungen....264
Quarzoszillator - Im DIP14 Gehäuse. 265
Über den Author....267

Vorwort

Wir leben in einer schnelllebigen Zeit. Die technische Entwicklung schreitet mit großen Schritten voran. Wer erinnert sich heute noch an Musikkasetten oder Schallplatten? Heute, im Zeitalter des Internetradios, MP3 und Blu-ray. Schnell geraten heute obsolete, technische Geräte in Vergessenheit.

Und doch sind diese Geräte und die damaligen Entwicklungen die Wurzeln der heutigen Technik.

Dem ‚Vergessen' entgegen zu wirken, haben sich nicht nur Museen verschrieben, sondern kleinere Gruppen – meist über das Internet verbunden – halten die damalige Technik am ‚Leben'.

In diesem Buch geht es im Besonderen um C64 und Amiga 500 - Homecomputer der 80/90er Jahre. Dabei soll es eben kein Buch über die Historie dieser Computer ‚for the masses' werden, davon gibt es schon einige, exzellente Bücher am Markt, sondern es soll im speziellen auf einzelne, aktuelle Entwicklungen zu diesen Systemen hingewiesen werden. Auch die Restauration solcher ‚antiker' Rechentechnik, soll hier und da aufgezeigt werden. Deshalb auch eine Bitte. Selbst defekte Geräte haben als Ersatzteilträger einen ‚Wert'. Nicht wegwerfen, sondern den entsprechenden Gruppen zugänglich machen!

Ich möchte Projekte und Kontaktadressen aufzeigen. Evtl. wird ja dadurch bei den einen oder anderen Lesern die Leidenschaft (wieder) erweckt, sich mit diesen Themen zu beschäftigen.

Durch die oben genannte Schnelllebigkeit, können Informationen inzwischen schon wieder veraltet, Internet-Links nicht mehr gültig sein. Dann sollte meistens eine der aktuellen Internetsuchmaschinen helfen können, die entsprechenden Informationen und Websites wiederzufinden.

Alle hier aufgeführten Verfahren, Bauvorschläge etc. sind nach besten Gewissen – zum Teil selbst nachgebaut worden. Eine Gewährleistung kann aber in keinerlei Hinsicht gegeben werden. Ein Nachbau und Betrieb erfolgt auf eigene Gefahr! Es sind die üblichen Kenntnisse im Umgang mit den entsprechenden Werkzeugen und Materialien zu beachten.

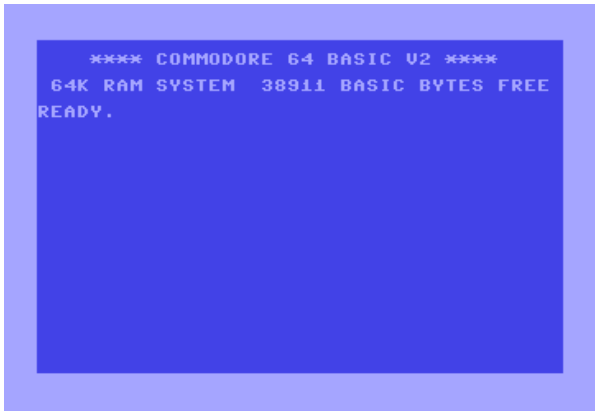
Der C64

Der Commodore 64 (kurz: C64, umgangssprachlich auch 64er) ist ein 8-Bit-Heimcomputer mit 64 KB (Kilobytes = 64 x 1024 Bytes) Arbeitsspeicher.

Äußerst populär war der von Commodore gebaute C64 Mitte bis Ende der 80er Jahre sowohl als Spielkonsole als auch zur Softwareentwicklung. Er gilt mit über 17 Mio. verkauften Geräten weltweit als der meistverkaufte Heimcomputer aller Zeiten.

Im Gegensatz zu modernen PCs verfügte der C64 über keine internen Massenspeichergeräte. Alle Programme mußten von externen Laufwerken, wie dem Kassettenlaufwerk Datasette oder dem 5¼″-Diskettenlaufwerk 1541, geladen werden.

Nach dem Einschalten wurde man mit einem sog. Basicprompt begrüßt.



Der C64 bot viele Schnittstellen und war daher auch für Bastler und Hardwaretüftler interessant. Für die Spieleentwicklung waren die - für damalige Verhältnisse – gute Grafikfähigkeiten und der auch heute noch großartige Sound über den SID Chip verantwortlich.

Durch die sehr große, gebaute Stückzahl, ist ein C64 System auch heute noch, über einschlägige Online-Auktionshäuser und Gebrauchtmärkte günstig zu bekommen und eignet sich daher ideal als Bastler, Restaurationsobjekt, da ggf. bei Fehlschlägen, sich finanzielle Verluste in Grenzen halten.

Kleine C64 Reparaturanleitung

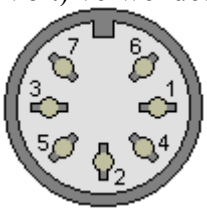
Benötigtes Werkzeug

Schraubendreher
Digitalmultimeter
funktionierendes Spielmodul
Logiktester (High/Low/Pulse) oder Oszilloskop
Entlötwerkzeug (Pumpe+Litze+Kanüle)
Lötstation geregelt
IC-Sockel
getestete Ersatz-Chips (*)
(*) Ohne solche getesteten Ersatzchips kann man es vergessen! Dazu benötigt man einen C64, wo allen großen IC's und mind. 1 Ram gesockelt sind. Dann können ausgelötete Chips hier getestet werden...

Wenn der C64 nicht mehr funktioniert.

1. Zuerst einmal alles abziehen (IMMER im stromlosen Zustand!!). Floppykabel, Module, Resettaster extern, Datasette, Joysticks, Maus etc. Auch das Netzteil.. das sollten wir erst einmal separat prüfen.

2. Netzteil prüfen. Das Netzteil liefert 5 Volt Gleichspannung (DC) und 9 Volt Wechselspannung (AC). Die 5 Volt des externen Netzteils sollten mit einem Strom von mindestens 1,0 Ampere belastet werden können. Die 9 Volt Wechselspannung dienen für die interne Uhr als Referenzfrequenz, und werden im C64 nach Gleichrichtung auch für die Spannungsversorgung des SID-Chips (9 oder 12 Volt) und des Motors für die Datasette (6 Volt) verwendet. Belegung der Buchse am C64:



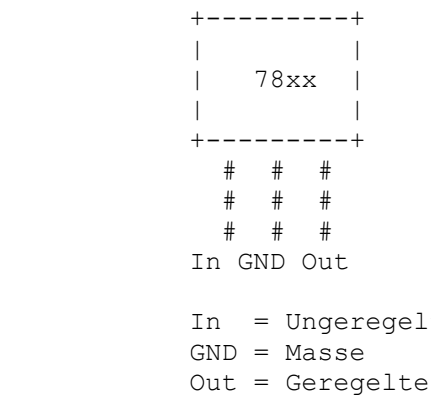
Pin	Signal	Bemerkung
1	GND	Masse
2	GND	Masse
3	GND	Masse
4	-	unbelegt, evtl. auch 5 Volt
5	+ 5V DC	5 Volt Gleichspannung
6	9V AC	9 Volt Wechselspannung
7	9V AC	9 Volt Wechselspannung

Hier also erst einmal die Spannungen des Netzteils prüfen. Die 5V sollte so im Bereich 4,95 bis 5,05 sein. Evtl. sonst die Sicherung (schwarze Schraubverbindung) prüfen und ggf. ersetzen 250V/Träge/200mA (C64-I = 150mA).

3. Wenn das Netzteil ok ist, kann man es mit dem C64 verbinden. Für die Überprüfung muß der C64 geöffnet werden. Beim C64-I gibt es an der Gehäuseunterseite vorne drei Schrauben, die gelöst werden müssen. Danach kann das Gehäuse einfach nach oben aufgeklappt werden. Man findet eine Abschirmpappe, die mit Alufolie bezogen ist. Diese Pappe ist hinten rechts an einem Abschirmblech festgeklemmt. Dort kann man die Klammer einfach nach vorne

abziehen und die Pappe nach oben wegklappen. Nun wird der Blick auf die Sicherung an der rechten Seite des Boards frei. Der C64-C wird durch drei Kunststoffklammern an der Unterseite des Gehäuses zusammengehalten.

Diese kann man einfach mit einem Schlitzschraubendreher auseinander biegen und dabei das Gehäuse aufziehen. Nachdem die Gehäuseoberhälfte nach oben entfernt wurde (Tastatur+LED einfach am Stecker abziehen), muß nun wieder die Abschirmpappe weggeklappt werden (Einige sind auch der Meinung, die sollte man ganz weglassen, um die Wärmeabfuhr zu fördern). Darunter befindet sich das Board mit der Sicherung auf der rechten Seite. Diese Sicherungen sind größer als die im Netzteil! Daten: 250V / Träge / 1,5A Jetzt zuerst die große Sicherung herausnehmen und mit dem Multimeter auf Durchgang prüfen (Die Betriebs-LED leuchtet übrigens auch bei defekter interner Sicherung des C64!). Falls defekt, eine Neue einsetzen! Jetzt einschalten und die Spannungen kontrollieren. Es sind auf dem C64 Board noch 2 sog. 78xx Festspannungsregler verbaut. 1x 7805=5V und 1x7812=12V. Die Eingangsspannung sollte mind. 2,5V höher als die gewünschte Ausgangsspannung sein.



In = Ungeregelter Eingang +
GND = Masse
Out = Geregelter Ausgang +

Sollte die Spannungen nicht korrekt sein, so sind die entsprechenden Elkos und die Regler zu tauschen. (Auch hier sind einige der Meinung, die Elkos+Regler sollte man in jedem Fall mal nach 15-20J tauschen. Elkos altern!)

Auch an Pin 6=+5V und Pin 21=GND der CPU 6510 messen.

Dann noch mal über Antennenanschluß/Monitor prüfen, ob jetzt ein Basic Einschaltbild kommt.

Ab hier wird es komplizierter

Ein weiterer Test ist das blinde Eintippen von Floppy Befehlen. Dazu Floppy verbinden und einschalten. Die Floppy sollte einen Einschalttest durchführen. Dabei leuchtet die Power-LED (grün) ständig, die rote LED leuchtet für 2-3 Sekunden und dabei läuft der Floppymotor. Danach sind Floppymotor+rote LED wieder aus. Jetzt den C64 einschalten. Die Floppy bekommt dabei ein Reset-Signal und wiederholt obige Prozedur. Wenn bis hierhin ok, jetzt blind z.B LOAD "$",8 eintippen und Return drücken. Läuft die Floppy an und arbeitet für ein paar Sekunden? Wenn ja, könnte beim C64 'nur' die Bildwiedergabe gestört sein. Kabel zum Monitor/Fernseher prüfen, ggf. gegen ein getestetes Kabel tauschen. Kanal am Fernseher korrekt eingestellt? ggf. Suchlauf starten.

Ansonsten..

So. Sicherungen sind alle ok. Spannungen sind alle ok. Aber C64 tut es trotzdem nicht.

Falls nur SID+VIC gesockelt sein sollten, kommt man um ein Auslöten und Sockeln der IC's nicht herum! Dazu sollte man sich erst einmal an einer defekten Platine im Auslöten üben! Man sollte einen funktionierenden C64 haben, bei dem alle großen IC's und am besten auch noch ein RAM gesockelt ist. Alle ausgelöteten IC's sollten darin getestet werden und falls sie defekt sind, bitte sofort aussortieren!

Zum Thema auslöten. Ich verwende eine starke Entlötpumpe. Platine mit Lötseite nach oben auf den Arbeitstisch legen und fixieren. Lötstelle erhitzen und wenn Lötzinn gut verflüssigt ist, Lötkolben schnell wegziehen, gespannte Entlötpumpe sofort auf dem Lötauge plazieren und Schuß. Sollte einmal der Pin nicht frei sein, besser zuerst wieder etwas frisches Lötzinn nachführen und dann sofort erneut absaugen. Auch gut hat es sich bewährt, eine nicht freie Lötstelle (ohne Pin) zu erhitzen und dabei gleichzeitig eine gelbe Kanüle (Einwegspritze; Apotheke) zu erwärmen. Dann mit der Kanüle das Lötloch durchstoßen.

Wichtig ist auch die Information, das ein defekter CIA/SID sehr wohl den C64 lahmlegen kann, das aber auch ohne diese 3 IC's eine Einschaltmeldung kommt! Also erst einmal weglassen, bis man eine Einschaltmeldung bekommt, dann eins nach dem anderen wieder einsetzen. Ich hoffe es muß an dieser Stelle nicht extra erwähnt werden, das Einsetzen/Entfernen von Bauteilen IMMER im stromlosen Zustand erfolgen MUSS!

Der VIC=Video-Chip ist unter der silbernen Abschirmung. Dieser ist fast immer gesockelt und hier kann man ein getestetes Exemplar probieren.

In 90% aller Fälle ist eine defekte PLA Schuld. Man kann also direkt diese jetzt austauschen (Auslöten, Sockel einlöten, Ersatz-PLA einstecken), oder weiter versuchen den Fehler einzugrenzen bzw. zu lokalisieren. Symthome können z.B sein: Schwarzer Bildschirm. Bildschirm komplett in einer Farbe.

Reset prüfen: An Pin 40 der CPU 6510 messen mit Logiktester. Beim Einschalten für 1-2 Sekunden=Low, danach High. Siehe Symtone und Ursachen.

Takt prüfen: An Pin 39 der CPU muß der Systemtakt anliegen (ca. 1MHz).

Ein Modul einstecken (immer im ausgeschalteten Zustand!) und einschalten. Kommt jetzt ein Bildschirm? Wenn ja, könnte das auf ein defektes Basic ROM hindeuten.

Werden Chips beim Betrieb sehr heiß (so daß man sich verbrennen kann) - insbesondere RAM Chips? Das könnte hier auf einen Defekt hindeuten. Falls man z.B eine Einschaltmeldung bekommt. mit 0 Bytes free oder ähnlich deutet das auf defektes Ram hin. Defekte Ram's kann man lokalisieren, indem man intaktes Ram per Huckepack aufsetzt(=auf den eingelöteten Chip einfach einen intakten aufsetzen, sodaß alle Pin's guten Kontakt haben. Ggf. die Pin's des aufzusetzenden Ram's zuerst etwas zusammenbiegen). Kommt dann eine Einschaltmeldung? Aber nicht nur einen nach dem anderen Ram so testen, denn es könnten ja mehr als ein Chip defekt sein. Also am besten alle 8 Ram's vom Typ 4164 per Huckepack Methode überbrücken.

Ansonsten würde ich jetzt weiter sockeln und testen in dieser Reihenfolge:
Kernal-ROM
Basic-ROM
Char-ROM
CIA 1
CIA 2
Rams
TTL's

Symthome und Ursachen

C64 arbeitet soweit, aber Zugriff auf Floppy geht nicht? Könnte der 7406 invertierende Bustreiber und/oder falls vorhanden die Schutzdioden defekt sein.

C64 hat keine Farb- sondern nur S/W Darstellung. Ist es ein NTSC Gerät? (VIC+Quartz prüfen) Ansonsten könnte das 8701 Clock-IC defekt sein (neben VIC).

Kommen nach dem Einschaltet einfach Zeichen, so als würde jemand die Tastatur betätigen? Evtl. mal Joystick rausziehen oder AutoFire ausschalten. Ansonsten, CIA1 (neben Tastatur Stecker) prüfen.

C64 zeigt Einschaltbild mit Rahmen und richtigen Farben, aber keine Schrift und prompt. Ein Spielmodul funktioniert einwandfrei. Könnte das Basic ROM defekt sein.

Wenn das Resetsignal an Pin 40 der CPU nicht ok ist, dann kann etwas mit U20(NE556) und U8(7406) falsch sein. An Pin 9 des NE556 müsste das Resetsignal invertiert anliegen. Danach geht das Signal zu Pin 9 vom 7406, wo es invertiert und verstärkt wird, und dann der Resetleitung zugeführt wird.

Symtom: Blinkende Farben/Zeichen

Test:
```
10 for x=0 to 15
20 poke 55296, x
30 print x, " = ", peek(55296) and 15
40 next x
```

Sollte:
```
0 = 0
1 = 1
2 = 2
...
15 = 15
```

ergeben. Wenn nicht ist ggf. das Farbram (2114) defekt.

Symtom: Nur "2047 (..weniger als 38911) Bytes Free"

Wahrscheinlich ein oder mehrere Ramchips defekt.

Test:
```
A=2048:POKE A,0:?PEEK(A):POKE A,255:?PEEK(A)
```

Das Bedeutet der Wert $0/$FF (%11111111) wurde in den Speicher in Adresse A geschrieben und danach wird z.B $F7 (%11110111) gelesen. In diesem Fall ist das Bit 3 nicht gesetzt und würde bedeuten das in einem C64 mit 8 RAM-Bausteinen das IC U10 defekt ist.

Tabelle:

C64 with 8 RAM chips: RAM chip type 64kbit x 1: e.g. 4164

bit	
0	U21
1	U9
2	U22
3	U10
4	U23
5	U11
6	U24
7	U12

C64 with 2 RAM chips: RAM chip type 64kbit x 4: e.g. 41464

bit	
0,1,2,3	U10
4,5,6,7	U9 / U11

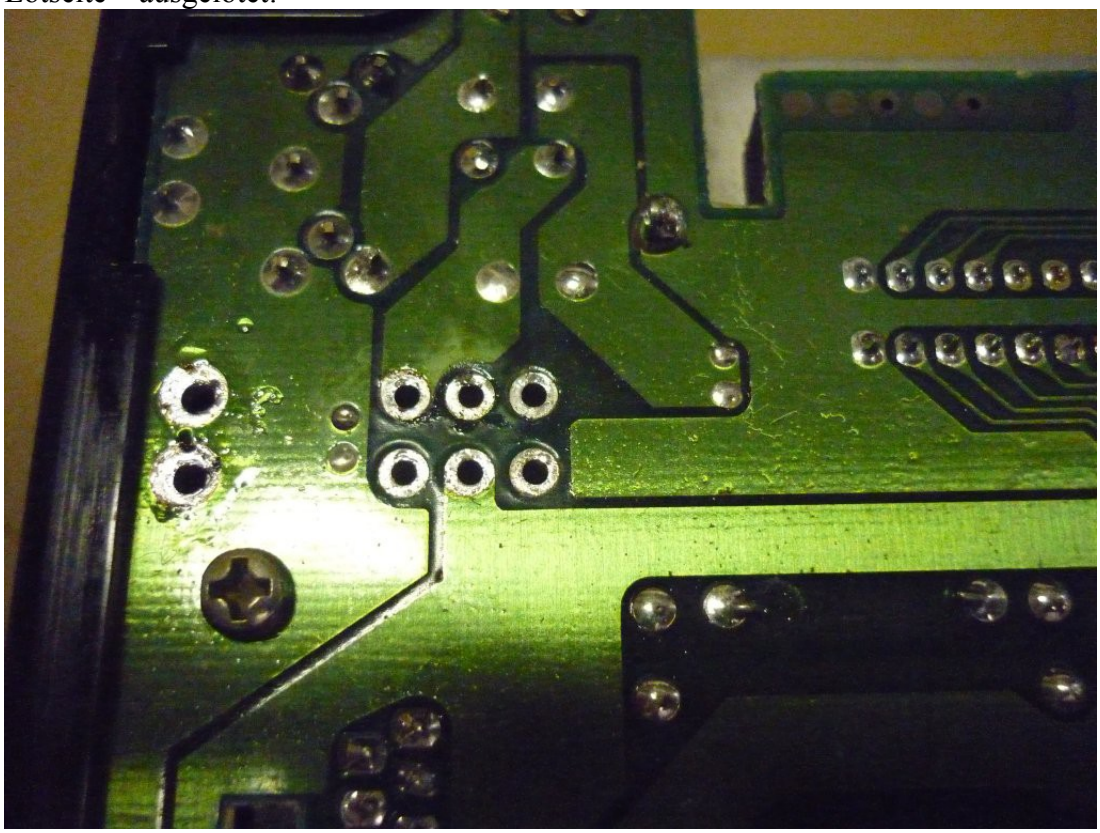
Wenn bei einem C64 (alte Boards mit 8xRAM) der obige Fehler auftritt, aber beim Einschalten immer mal unterschiedliche RAM-Angaben kommen, so kann es sein, dass der Multiplexer auf U13 (74LS257 / MOS 7708) defekt ist.

C64 Netzschalter Instandsetzung/Reparatur

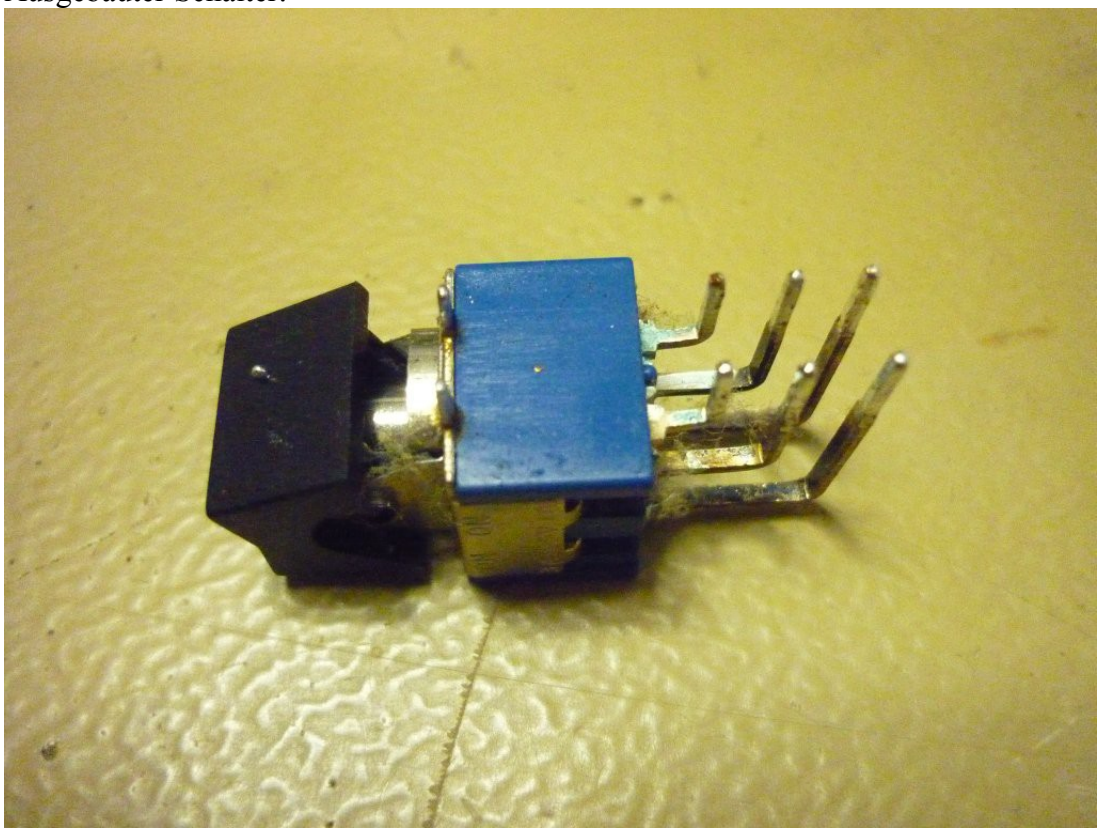
Von AREA51HD

Den Defekten Schalter auslöten:
Die Lötbohrungen sind sehr groß im verhältnis zu den anschlussbeinen des Schalters, das macht das auslöten mit Entlötpumpe, Litze oder Entlötstation recht einfach.
Das Entfernen des Schlotbleches erleichtert das Ausbauen des Schalters, ist aber nicht unbedingt notwendig. Die beiden Lötaugen bei der Massefläche sind die schwierigsten, und werden beim ersten Mal gerne vergessen.

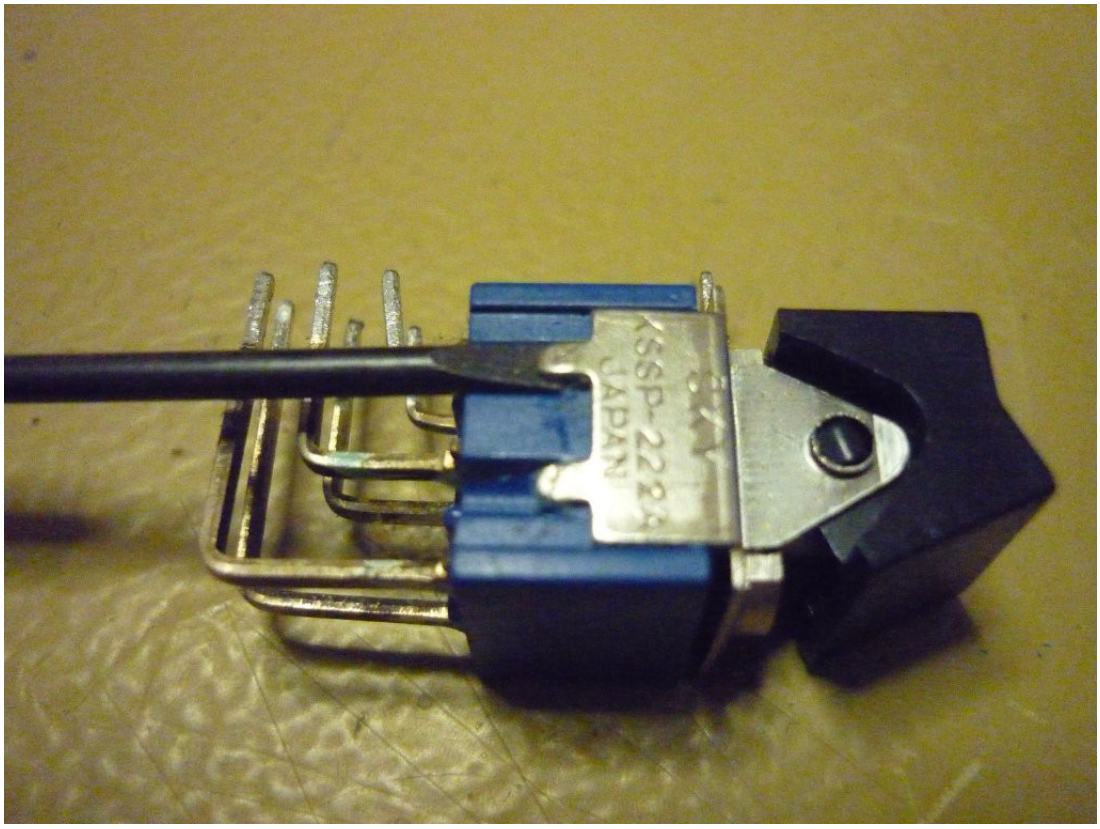
Lötseite – ausgelötet:



Ausgebauter Schalter:

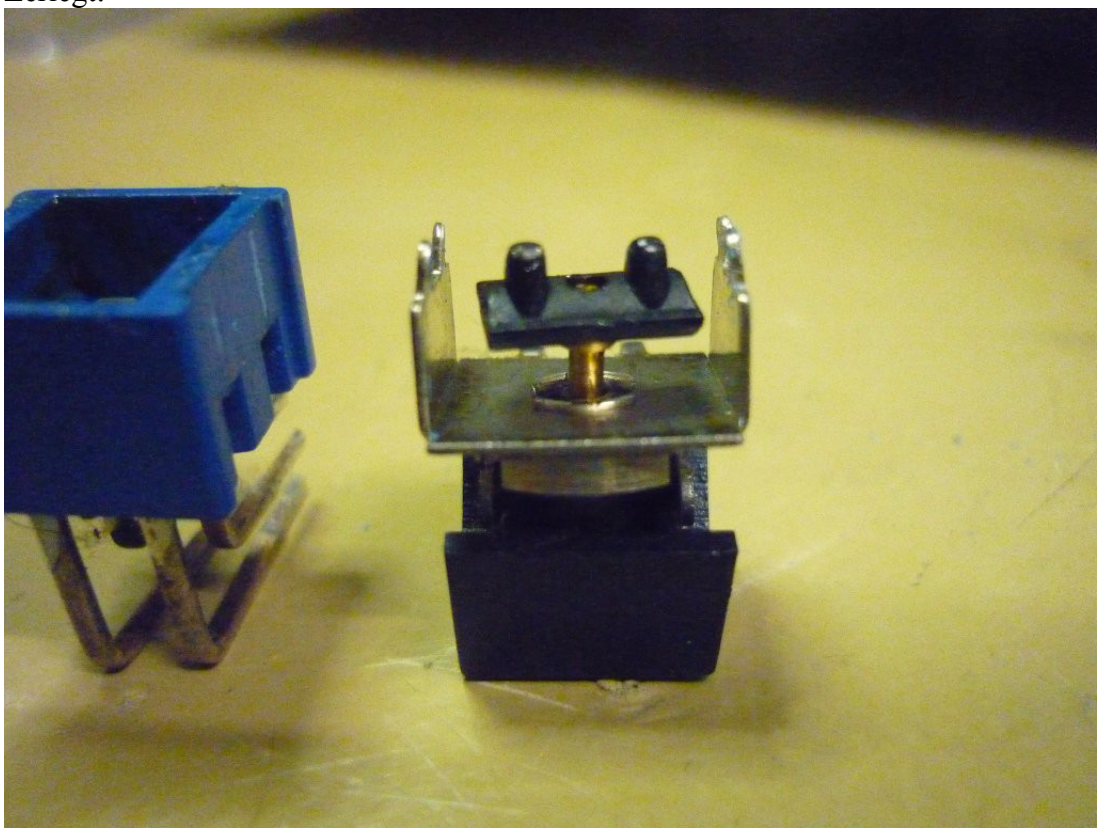


Links und Rechts am Schalter sind je zwei Laschen die den Schalter zusammenhalten.
Die Laschen vorsichtig mit einem kleinen Schraubendreher gradebiegen, und Langsam den Schalter auseinanderziehen. Achtung: Viele Kleinteile – Aufpassen das nichts verloren geht!

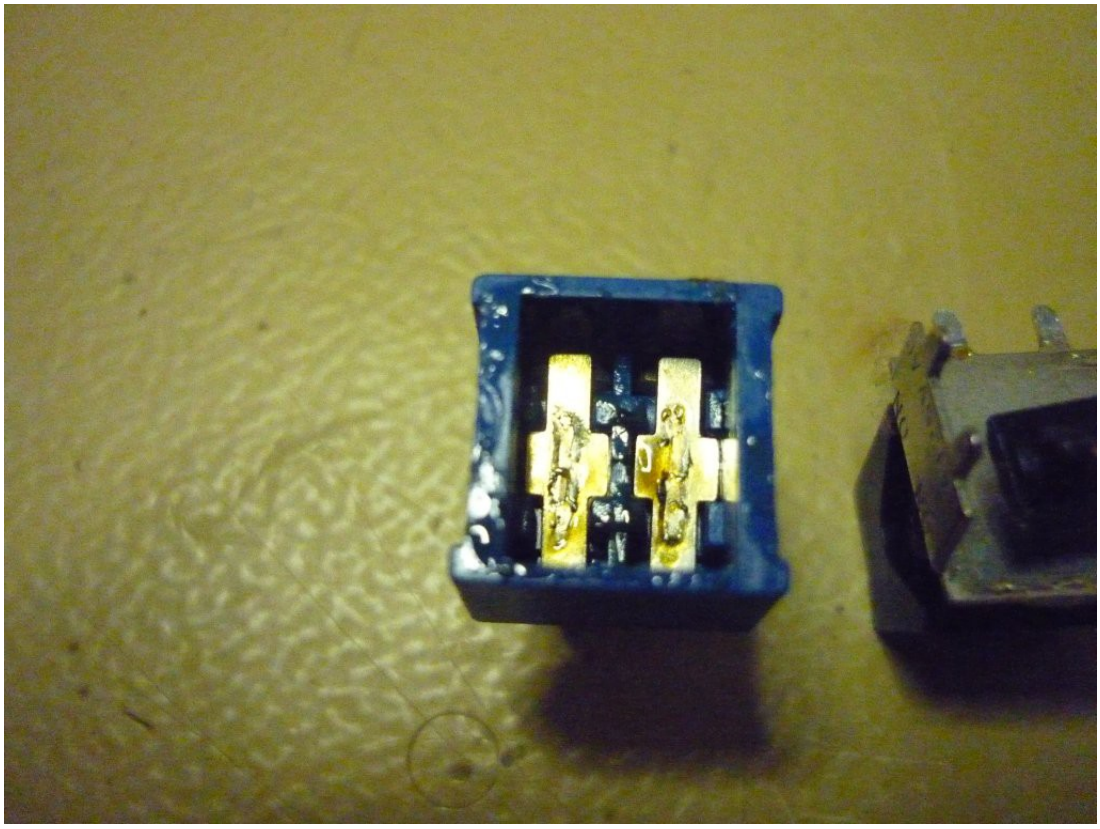


Auf dem Bild "Zerlegt" sieht man die Innerreihen richtig zusammengesteckt, dank der Feder fliegt das aber gerne auseinander.

Zerlegt:



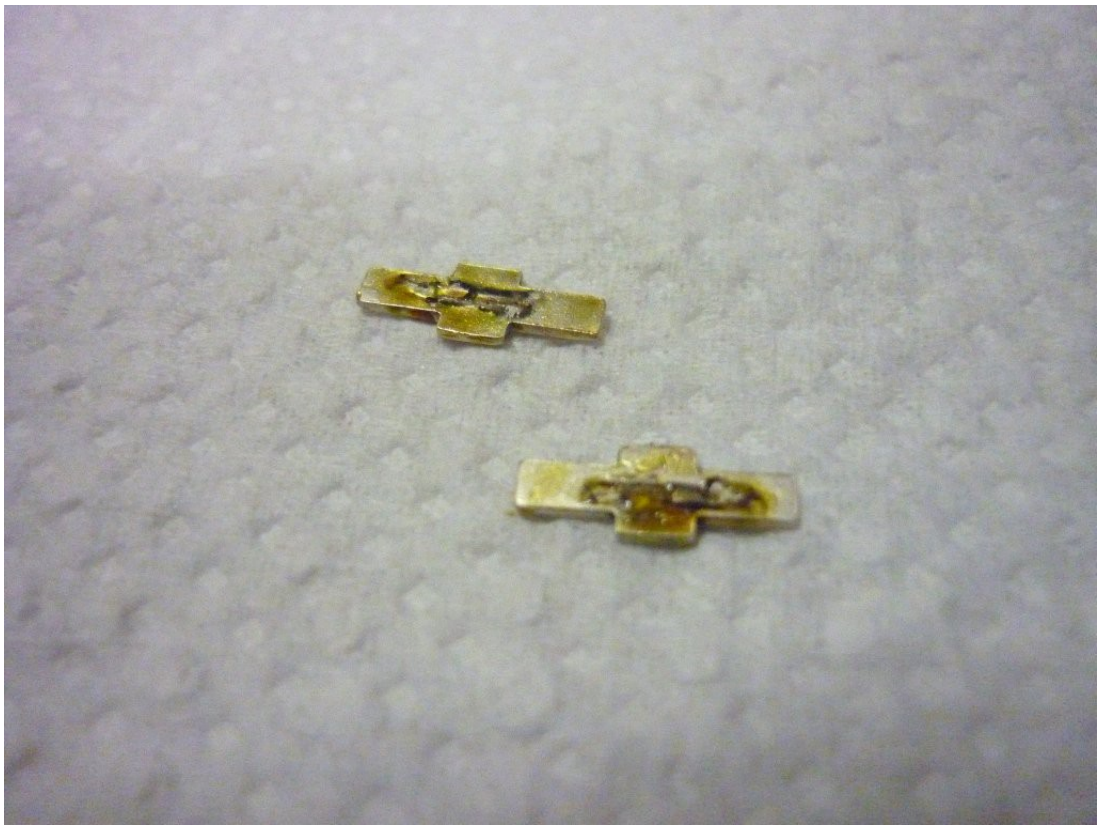
Kontakte mit Fett und Schmutz:



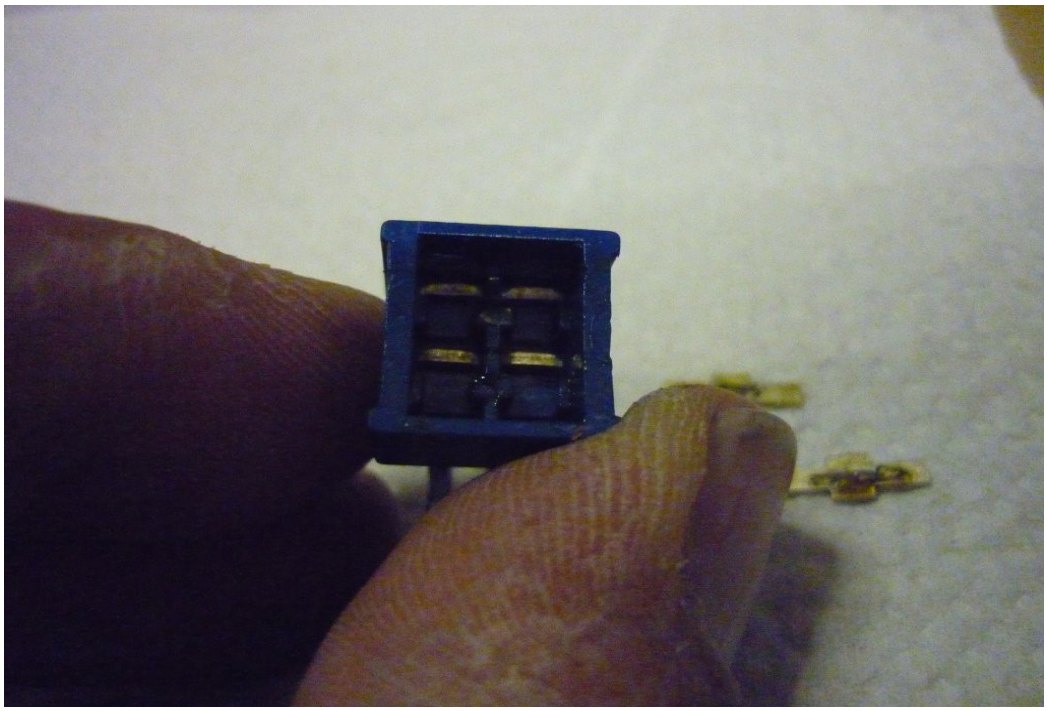
Die Schaltermechanik ist leider zu reichlich mit Fett behandelt worden. Nun nach Jahrzehnten wandert das Fett auf die Kontaktflächen und Stört das Schalten (Isoliert), oder das Fett kann Verharzen und den Schalter zum Klemmen bringen "Fettbernstein". Auch kann das Silber der Kontaktflächen anlaufen, und Silberoxid/sulfid isoliert.

Zuerst die Kontaktblättchen mit einem kleinen Schraubendreher rausheben, mit Glück reicht auch schon schütteln.

Kontaktblätter verschmutzt:

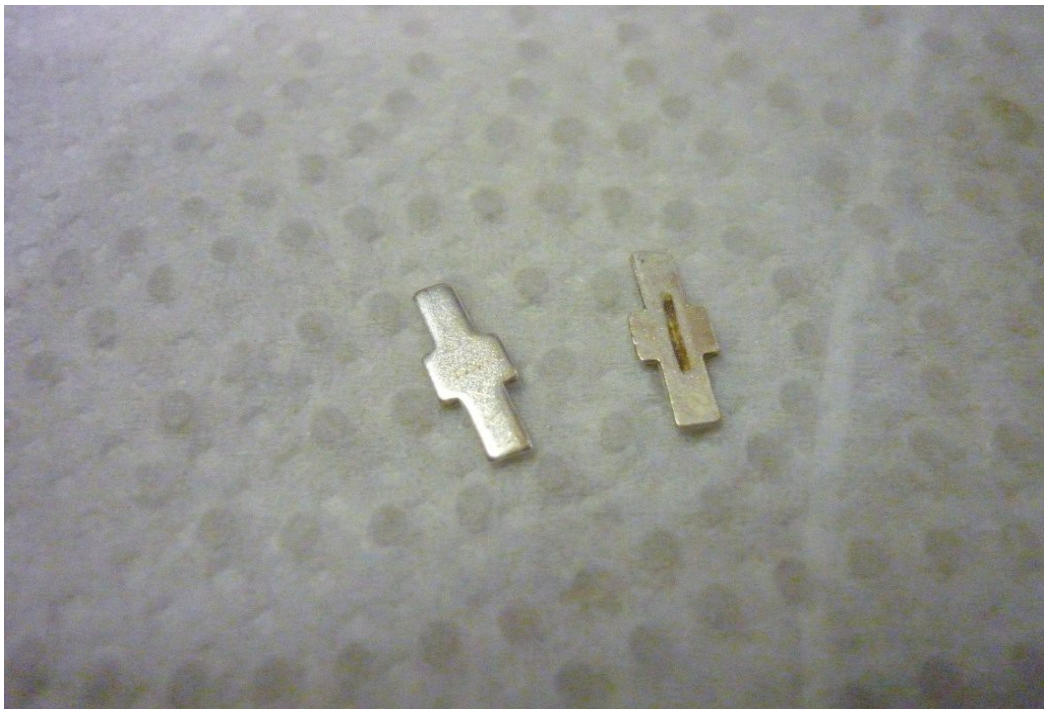


Auch die Gegenseite der Kontaktblättchen zeigt die Verschmutzung:

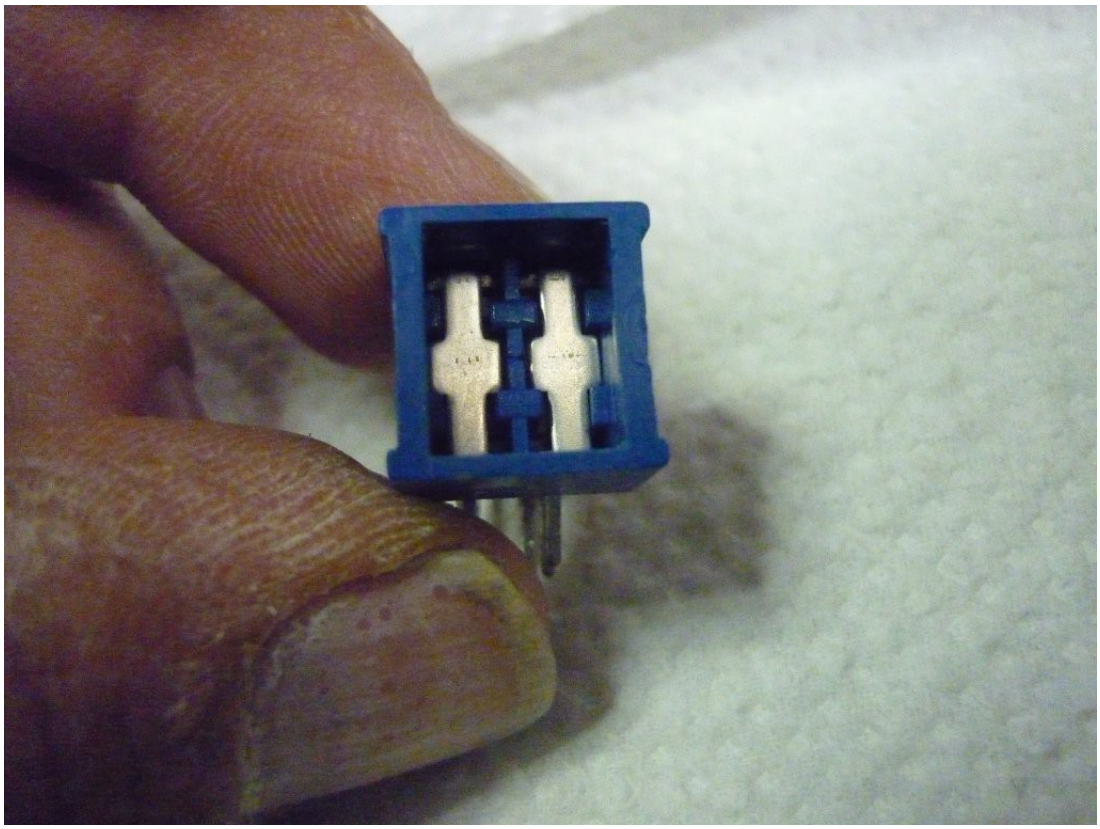


Ich benutze Bremsenreiniger aus dem Autohandel zum Entfernen des alten Fettes, Spiritus oder Isopropylalkohol gehen notfalls auch.

Dann Lege ich die Kontaktblätter auf ein Küchenpapier und gebe einen Schuss Kontakt60 Spray darauf. Mit Gummihandschuhen (Fingerschweiss kann Silberflächen wieder anlaufen lassen) reibe ich die Kontaktblättchen auf dem mit Kontakt 60 Getränkten Küchenpapier bis sie wieder schön sauber und blank sind.



Den Schalterunterteil fülle ich auch mit Kontakt 60 und lasse es einige Minuten einwirken, mit gleichzeitig vorsichtigen Bürsten der Kontaktflächen mit einem Pinsel.



Anschließend Wasche ich das Kontakt 60 mit Isopropanol weg, oder wer hat kann es auch mit Kontakt WL machen, wie der Hersteller CRC empfiehlt.

Was ich leider durch Schmerzen lernen musste, ist das Kontakt 60 ein Haltbarkeitsdatum hat! Danach wirkt die Rote Brühe nur noch Schlecht oder gar nicht mehr.

Und was noch Problematischer ist, die Dosen neigen dazu von Innen "Durchzurosten", und Laufen dann dort wo sie Stehen/Liegen aus. So habe ich mir ein Wohnzimmerregal versaut, und Flecken auf dem Teppich gemacht.

Die Kontaktblättchen auf Planheit prüfen, falls das Fett verharzt war und der Schalter mit Kraft Betätigt wurde, kann es die Kontaktblätter verborgen haben. Ggf. auf einer Metallplatte wieder gerade biegen.

Die Mechanik die am Schalteroberteil entfettet wurde (Den gemeinen Teil den die Feder zerlegt), mit einem winzigen Tropfen Vaseline einschmieren. Die Vaseline darf aber nicht auf die Kontaktflächen kommen.

Nun muss man alles zusammenfädeln.

Zuerst lege ich mit einer Pinzette die Kontaktblättchen wieder ein, ggf. kann man sie um 180° drehen um die unverbrauchte Kontaktseite zu nutzen.

Auch hier wieder darauf achten die Kontakte nicht mit den Fingern zu berühren (Pinzette und/oder Handschuhe).

Das Oberteil vorsichtig in das Unterteil bei leichter Schräglage einfädeln.

Wenn alles erst mal zusammengeschoben ist, den Schalter 2-3 mal betätigen.
Fehler beim Zusammenbau führen zum Klemmen oder fehlen der Rastpositionen.

Falls der Schalter normal zu Betätigen ist, weiter mit dem Laschen biegen, ansonsten wieder alles auseinanderziehen und neu probieren. Man braucht auch mit Übung häufig 2 Versuche bis es passt.

Die Laschen biegt man mit einem Schraubendreher runter, während eine Helfende dritte Hand den Schalter zusammendrückt.

Vor den Einbau prüft man nochmal den Schalter mit dem Multimeter, der Widerstand der Kontakte sollte im geschlossenen Zustand deutlich unter 0,5 Ohm sein.

Nun kann der Schalter wieder eingebaut werden, und sollte eine Weile halten.

C64 Ersatz Ein-/Ausschalter

Von Gerrit Heitsch

Bei Pollin unter der Nummer 420632 gibt es einen Ersatzschalter für einen unschlagbaren Preis (25 Cent).

Sieht auf dem Bild 1 nicht besonders passend aus, aber wie man auf den angehängten Bildern sieht ist die Wippe nur aufgeklipst, die gleiche Art der Befestigung der Wippe wie bei den Originalschaltern des C64. Mit einem sehr kleinen Schraubenzieher gab es kein Problem sie zu entfernen (Bild 2) und die Wippe eines defekten C64-Schalter aufzuklipsen. Das löst schonmal das erste Problem und man könnte ihn direkt in einem C16 oder Plus/4 verwenden.

Für die Verwendung in einem C64 ist etwas Bearbeitung der Anschlüsse nötig wie man auf dem dritten Bild sieht. Hierbei mit Gefühl arbeiten und sich etwas Zeit lassen. Drei davon kann man direkt passend biegen, der vierte braucht eine Verlängerung. Ich habe mir einen Draht des defekten Schalters geliehen. Die zwei übrigen Anschlüsse werden im C64 nicht verwendet, also habe ich sie einfach entfernt.

Ergebnis: C64 funktioniert wieder und ein sattes Schaltgefühl (Bild 3). Natürlich ist das ein Hack, aber besser als kein Schalter.

Müsste genauso mit den Schaltern einer 1541-II, eines C128 und einer 1581 funktionieren.

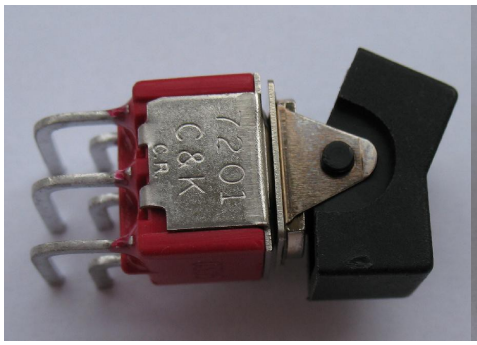


Bild 1

Bild 2

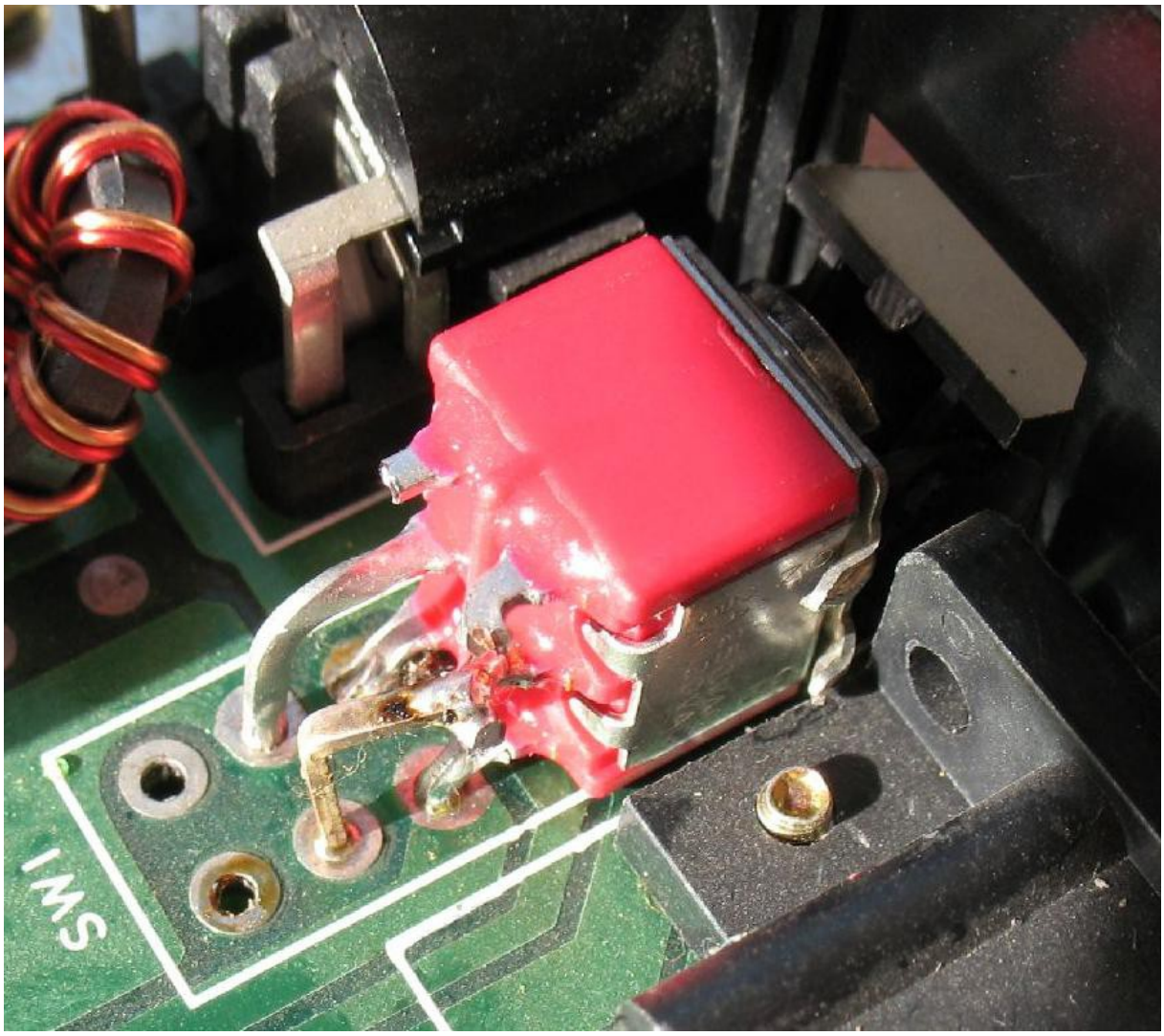


Bild 3

C64 Einbau Suppressordiode

Von Gerrit Heitsch

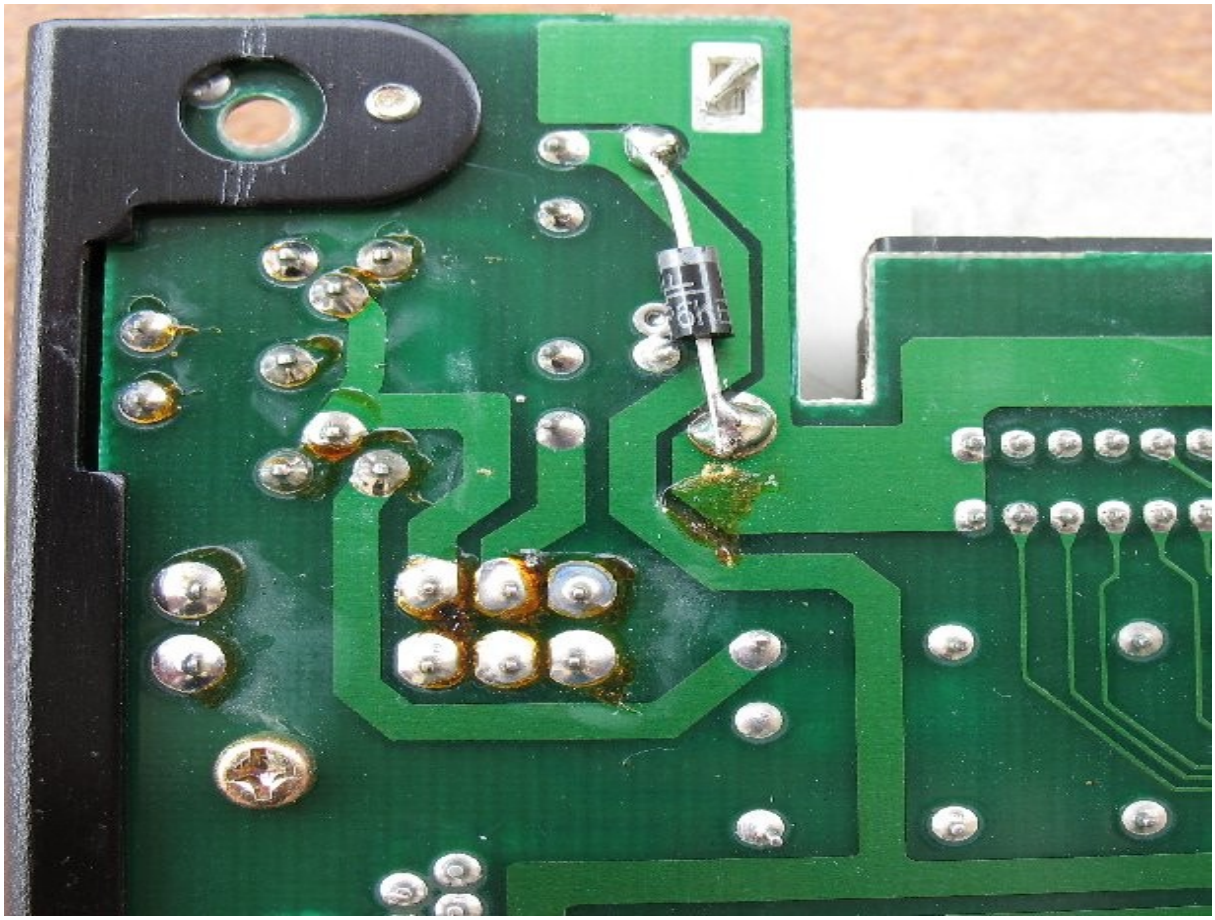
Suppressordioden werden beim Überschreiten einer bestimmten Spannung leitend und schneiden damit z.B. Peaks auf der Versorgung ab oder begrenzen so bei einem defekten Netzteil die Spannung bis sie selbst durchbrennen und zum kompletten Kurzschluss werden. In der Funktion wie eine Z-Diode, nur auf Schnelligkeit (< 1ns) und Ableitfähigkeit (> 50A) optimiert.

Die P6KE 6.8A beginnt ab 6.8V leitend zu werden. Nachdem die meisten ICs im C64 bis 7V zugelassen sind und bei kurzen Peaks deutlich mehr aushalten liegt die genau im Rahmen. Kostet keine 20 Cent das Stück. (Reichelt: P6KE 6,8A).

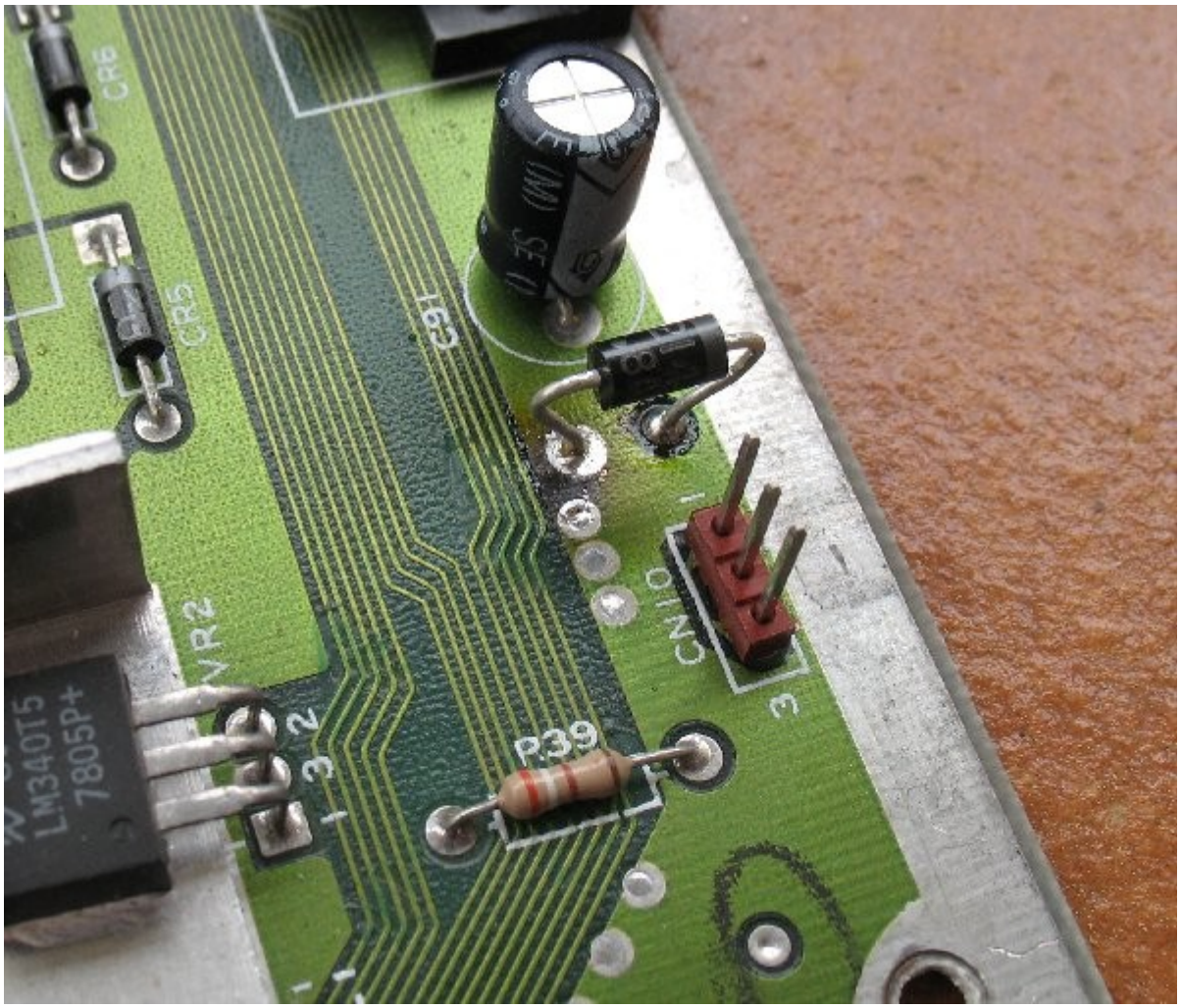
Damit keine Missverständnisse entstehen: So eine Diode ist kein absoluter Schutz, sie erhöht nur die Chance, dass bei defektem Netzteil oder sonstigem was die Spannung auf mehr als 6V ansteigen lässt, der C64 diesen Event überlebt und entweder gar nichts oder nur die billige Diode defekt ist.

Die folgenden Fotos sollten ausreichen um den Platz identifizieren zu können.

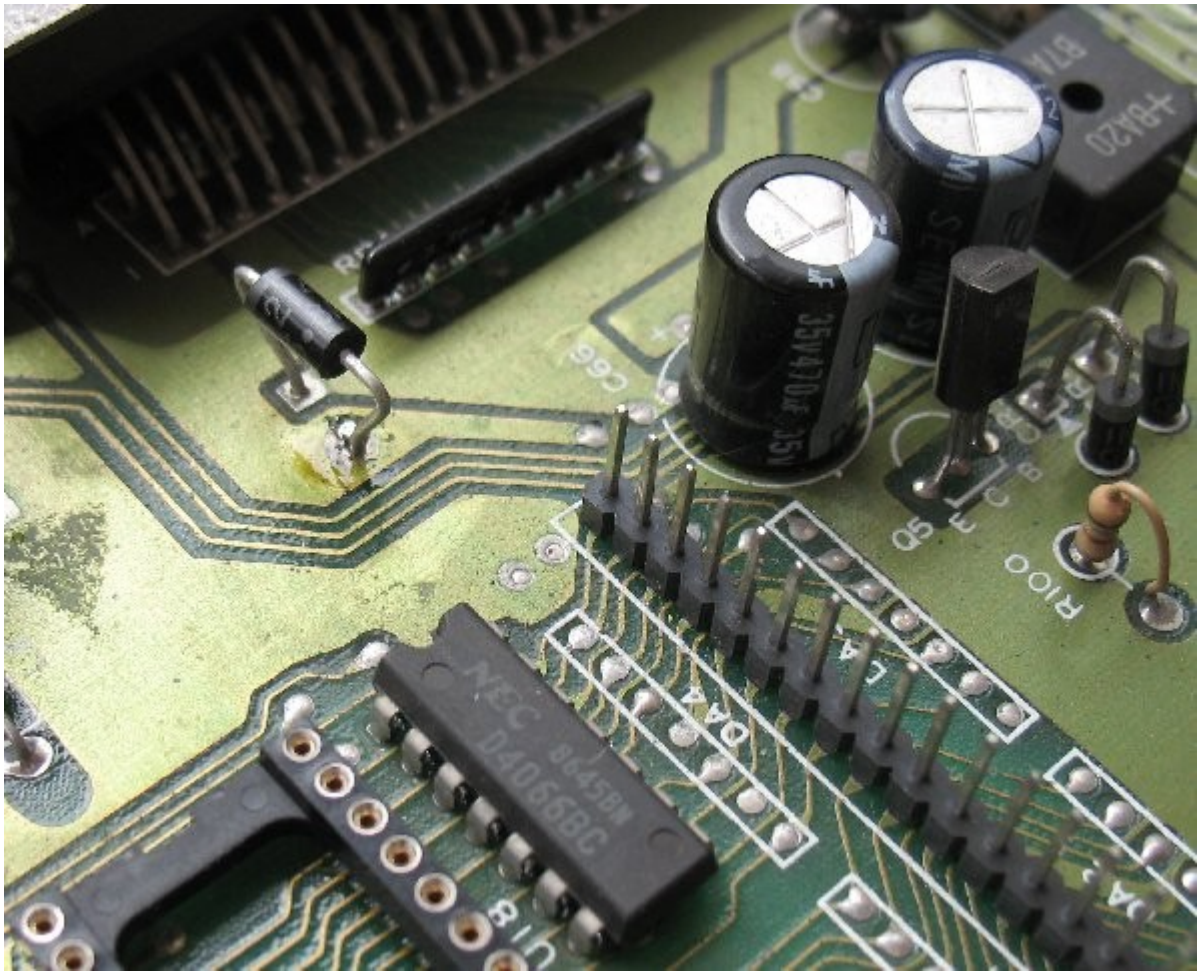
Hier zuerst ein Foto der 250407 (Rev.B), hier fand ich nur auf der Unterseite einen brauchbaren Platz für die Montage. Der Ring der Diode wird immer mit den +5V verbunden.



Hier die 250425 und 250466, der Einbauplatz ist identisch. Wegen der großen Massefläche ist ein starker Lötkolben nötig. Die Lötaugen sind vorhanden und müssen nur vom Zinn befreit werden. Wozu die gedacht waren ist unklar, Tests der Platine? Kondensator der wegoptimiert wurde?



Hier noch die 250469.



Es wurde immer darauf geachtet, daß die Diode mögilchst nahe an der Netzteilbuchse sitzt und die Leiterbahnen breit genug sind um im Zweifelsfalle etwas Leistung abzukönnen.

C64 Bausteine (Chips)

Platinen Versionen

Ur-C64 Platine aus 1982:

Sog. KU-Platine zählt auch noch zu den Ur-C64 Platinen:

Assy 250407 aus ca. 1983/4. PLA neben CPU, SID darunter:

Assy 250425 ab 1984. Achtung! Gegenüber 250407 sind PLA+SID an den Positionen getauscht! Taktteilung über 8701:

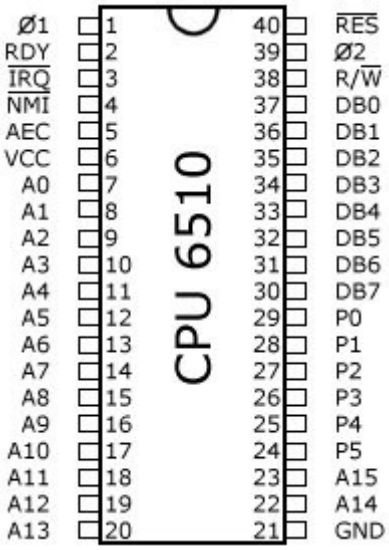
Assy 250466. Seltene Platinenversion. Sog. B3-Platine mit zusammen gefaßten RAM's:

Assy 250469. Letzte Platinenversion. Kernal+Basic in einem ROM:

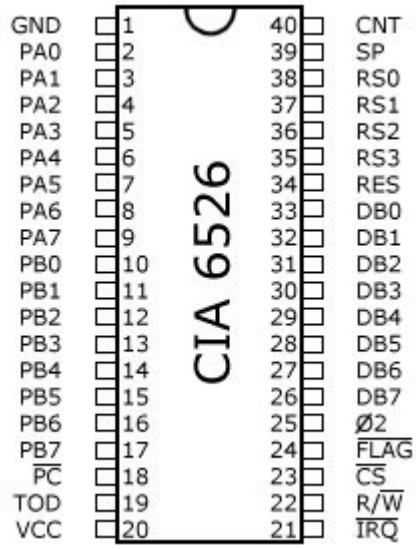
CBM Bezeichnungen:

```
901225    4k Char ROM
901226    8k Basic ROM
901227    8k Kernal ROM
906114    PLA-Ersatz durch 27512 Eprom+Adaptersockel möglich!
8701      Clock converter IC
4164      64kx1 DRAM (8 IC's im C64)
7707      7406
7708      74LS257
7709      74LS258
7711      74LS139
7712      74LS08
```

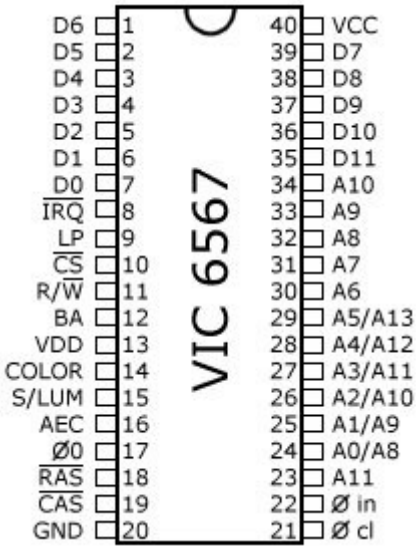
CPU 6510



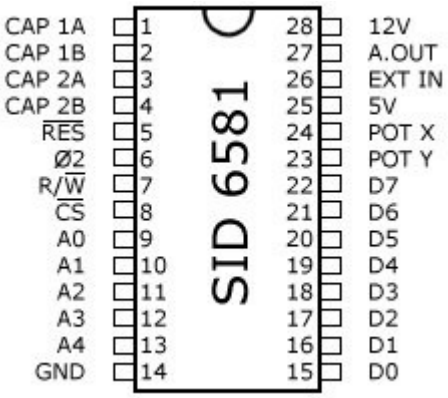
CIA 6526



VIC 6567/9



SID 6581



Der SID des C64-II = 8580 ist Pin kompatibel zum 6581. An PIN 28 sind dann aber nur 9V und nicht 12V anzulegen!

Wenn man einen 8580 dort betreiben möchte, wo eigentlich ein 6581 rein gehört, kann man einen ca. 137 Ohm Widerstand zw. PIN 28 des SID und PIN 28 der Fassung einfügen, um die 12V auf ca. 9V zu reduzieren. Dagegen kann ein 6581 problemlos in einem C64 C betrieben werden.

Hier gibt es ein Projekt, den SID durch einen AVR zu ersetzen:
http://www.swinkels.tvtom.pl/swinsid/

RAM

```
2114
1kx4 SRAM.

    +---+--+---+
 A6 |1  +--+ 18| VCC
 A5 |2       17| A7
 A4 |3       16| A8
 A3 |4       15| A9
 A0 |5  2114 14| D1
 A1 |6       13| D2
 A2 |7       12| D3
/CE |8       11| D4
GND |9       10| /WE
    +----------+
```

```
4116
16kx1 DRAM.
Requires -5, +5 and +12V.

     +---+--+---+
 VEE |1  +--+ 16| GND
   D |2       15| /CAS
 /WE |3       14| Q
/RAS |4  4116 13| A6
  A0 |5       12| A3
  A2 |6       11| A4
  A1 |7       10| A5
 VDD |8        9| VCC
     +----------+
```

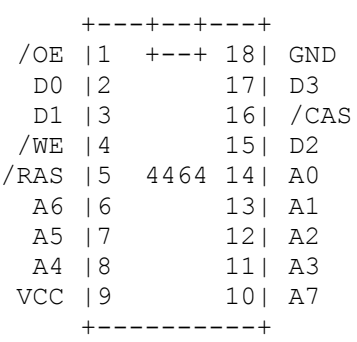
Verwendung eines 4164 Chips statt einem 4116 Chip

- Pin 1 und 8 hochbiegen, so dass sie nicht mehr in die Fassung passen.
- Pin 8 und 9 mit leitendem Draht verbinden.
- Pin 1 und 8 so abkneifen, das keine anderen Bauteile berührt werden können.
- Baustein dann so in die Fassung einsetzen.

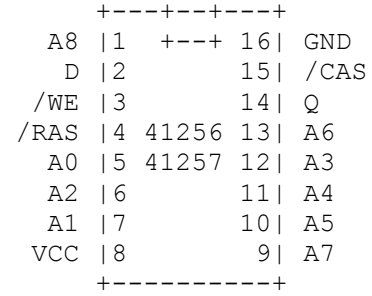
```
4164
64kx1 DRAM.

     +---+--+---+
     |1  +--+ 16| GND
   D |2       15| /CAS
 /WE |3       14| Q
/RAS |4  4164 13| A6
  A0 |5       12| A3
  A2 |6       11| A4
  A1 |7       10| A5
 VCC |8        9| A7
     +----------+
```

Anstatt eines 4164 kann auch ein 41256 verwendet werden.
Dazu einfach ein Kabel zwischen Pin 1 und 16 anlöten.

4464
64kx4 DRAM.



41256, 41257 256kx1 DRAM.



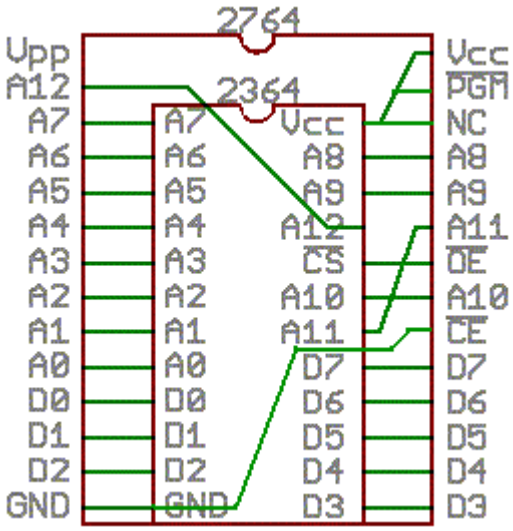
```
44256, 44258
256kx4 DRAM.

     +---+--+---+
  D0 |1  +--+ 20| GND
  D1 |2       19| D3
 /WE |3       18| D2
/RAS |4       17| /CAS
     |5 44256 16| /OE
  A0 |6 44258 15| A8
  A1 |7       14| A7
  A2 |8       13| A6
  A3 |9       12| A5
 VCC |10      11| A4
     +----------+
```

ROM

Der C64 (alte Bauform) hat 3 ROM's. U3 = 8k Basic Interpreter, U4 = 8k Kernal und U5 = Character ROM. Dabei handelt es sich um maskenprogrammierte Chips mit diesen MOS Nummern:

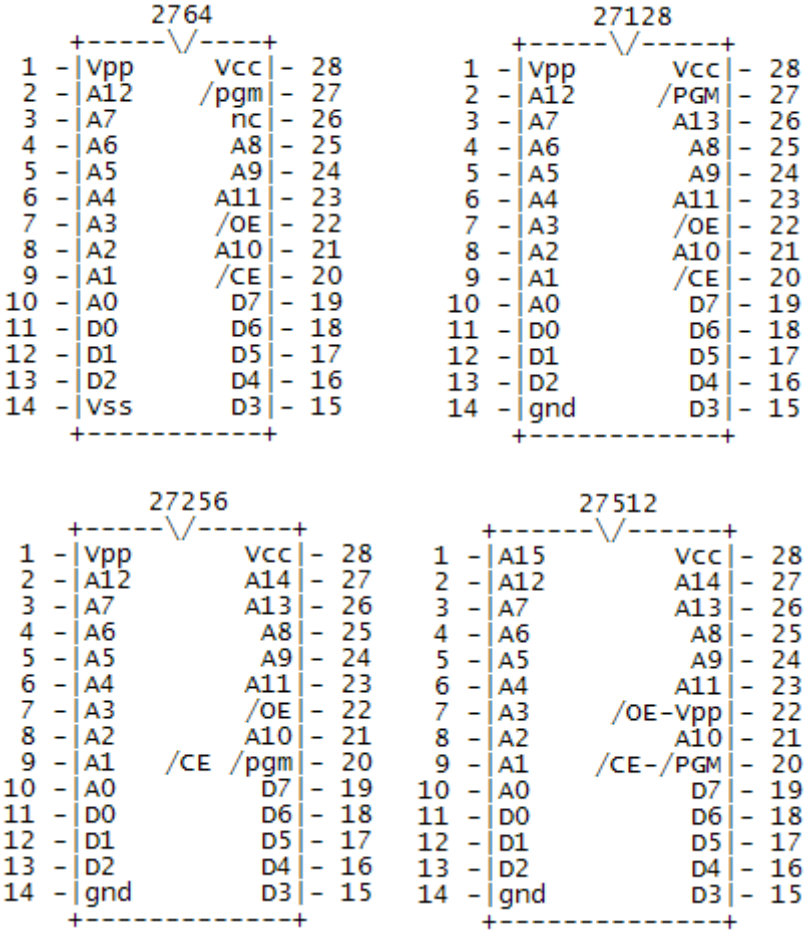
IC	MOS-part	Memory region	Standard PROM part	Description
U3	901226-01	0xA000-0xBFFF	2364A	BASIC ROM
U4	901227-03	0xE000-0xFFFF	2364A	Kernal ROM
U5	901225-01	0xD000-0xDFFF	2332A	Character ROM

Hier eine Adapterschaltung um Standard 2764 Eproms als ROM Ersatz zu verwenden:



Vom 2364 Pin 24=Vcc mit 2764 Pin 28,27,26 verbinden.
Vom 2364 Pin 21=A12 mit 2764 Pin 2 verbinden.
Vom 2364 Pin 18=A11 mit 2764 Pin 23 verbinden.
Pin 20 am 2764 /CE mit GND verbinden (Pin 14 am 2764).

Eprom Pinouts



Messpunkte mit dem Oszi beim C64

Von Gerrit Heitsch.

Dieses Kapitel ist nicht als Einstieg bei der Fehlersuche gedacht. Es wird folgendes vorausgesetzt:

1. Es wurde bereits getestet ob irgendwelche ICs ungebührlich heiss werden (= man verbrennt sich die Finger)
2. Die Versorgungsspannungen wurden gemessen
3. Wenn möglich wurden alle gesockelten ICs in einem funktionierenden Board getestet.

Es wird ausserdem vorausgesetzt, dass Grundlagen des Umganges mit dem Oszi vorhanden sind (man sollte die Anleitung gelesen und zumindest teilweise verstanden haben) und bekannt ist, wie man bei einem DIP-IC die Pins zählt. Idealerweise hat man ein funktionierendes Board an dem man Vergleichsmessungen machen kann.

Das von mir verwendete Oszi ist ein UNI-T UTD2102CEL, mehr als ausreichend für den Hobbybereich.

Für Messungen in alten Rechnern mit Takten unterhalb von 10MHz benutze ich meist folgende Einstellungen:

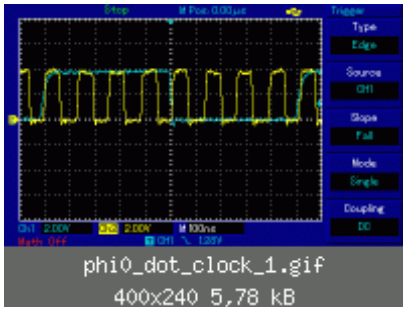
- Vertikal: 2V, damit ergeben 5V 2,5 Kästchen auf dem Bildschirm. Bei NMOS erreicht HIGH normalerweise nur Werte um ca. 4V.
- Horizontal: 100ns, damit hat man eine gute Auflösung von ca. je 5 Kästchen für HIGH und LOW bei PHI0 und sieht auch Störungen gut.
- Triggerung auf fallende Flanke in der Mitte des Bildschirms, damit ist der CPU-Teil eines PHI0-Zyklus auf der linken Seite des Bildschirms und der VIC-Zyklus auf der rechten Seite zu sehen.
- Das Oszi ist ein Zweikanal-Oszi und ich stelle die meisten Messungen im Verhältnis zu PHI0 dar wobei die Triggerung auf PHI0 passiert. Damit steht das Signal nicht leer im Raum sondern man sieht den Kontext zum Rest des Systems
- Manche Messungen brauchen für ein gutes Bild ein Speicheroszi, Einstellung auf 'one shot' oder 'single' (also nur die erste Triggerung auswerten und darstellen) und mehrfache Starts der Messungen bis das erwartete Bild zu sehen ist.

Die folgenden Einträge geben Hinweise welches Signal wo zu finden ist und wie es auf einem funktionierenden Board aussehen sollte, teilweise mit Bildern.

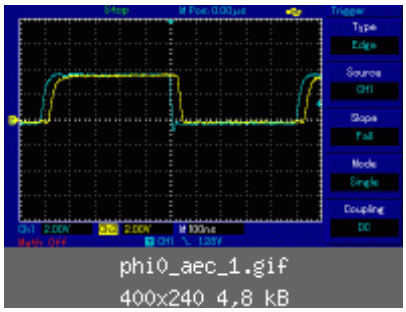
Als erstes verifizieren ob das RESET-Signal sauber erzeugt wird. Es ist an der CPU auf Pin 40 zu finden und muss beim Einschalten LOW (0V) sein und nach kurzer Zeit auf HIGH (4-5V) wechseln und dort bleiben. Passiert das nicht läuft die CPU nicht los. Mögliche Fehlerquellen sind hier der 556 und der 7406, bei einem 250469 auch der 74LS14.

Der nächste Check: Bekommt der VIC auch seine Takte? Diese liegen an Pin 21 (color clock, 17,73 MHz) und Pin 22 (dot clock, 7,88MHz) an. Das gesamte Systemtiming wird aus 'dot clock' abgeleitet, fehlt dieser Takt geht gar nichts. PHI0 (Pin 1 der CPU) ist genau 1/8 davon.

Angehängt ein Bild, PHI0 in Relation zur dot clock, aufgenommen auf einer 250466 mit 6569R5. PHI0 ist bei solchen Messungen hellblau dargestellt, das interessante Signal ist gelb:



Als nächstes wird AEC verifiziert (Pin 5 der CPU) in Relation zu PHI0. Wenn man nicht gerade eine Badline erwischt sieht es aus als wenn AEC dasselbe ist wie PHI0, nur leicht verschoben. Bei einer Badline wäre AEC die ganze Zeit LOW:



Jetzt die Steuerung der RAMs. Die Schaltung im C64 routet _RAS vom VIC direkt zu den RAMs während _CAS durch die PLA muss. Ohne _CAS rückt das RAM keine Daten raus und macht nur einen Refreshzyklus mit der angelegten Zeilenadresse. Das Routing durch die PLA erlaubt es auf diese einfache Weise das RAM je nach Anforderung abzuschalten. Es verzögert aber auch _CAS etwas.

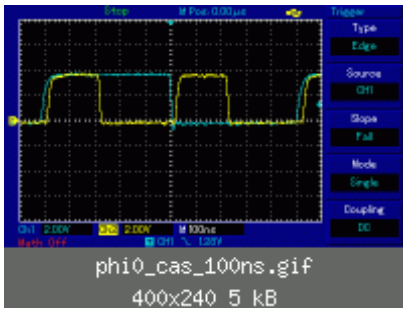
Der normale Ablauf ist das zuerst _RAS auf LOW geht und kurze Zeit später _CAS. Am Ende des Zyklus sind beide wieder HIGH und bleiben das auch eine Weile um die Precharge-Time einhalten zu können.

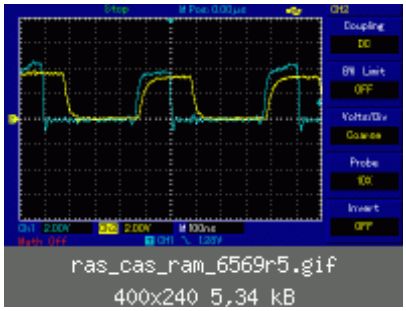
Die folgenden Bilder zeigen _RAS und _CAS im Verhältnis zu PHIO und je einmal im Verhältnis zueinander am VIC und am RAM. Bei den letzteren ist der Trigger auf _RAS welches hellblau dargestellt wird.

Die Pins sind:

- _RAS am VIC: 18
- _CAS am VIC: 19
- _RAS am 4164: 4
- _CAS am 4164: 15
- _RAS am 41464: 5
- _CAS am 41464: 16

Bis zum 250425 benutzt der C64 8 Stück 4164 (64k x 1 bit), ab dem 250466 dann 2 Stück 41464 (64k x 4 bit) DRAMs. Teilweise steht eine andere Nummer drauf, aber die Pinbelegung ist dieselbe.



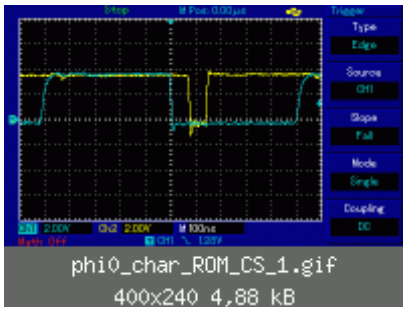


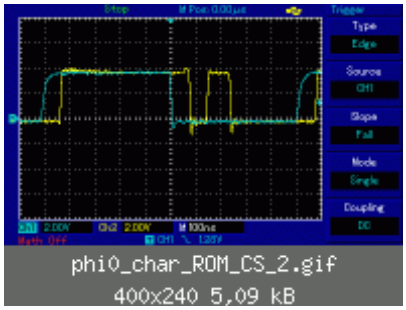
Die nächste Check ist ob die ROMs überhaupt angesprochen werden. Dazu einfach das _CS-Signal an Pin 20 des ROMs testen. Es darf nicht dauernd HIGH sein sondern muss hin und wieder nach LOW wechseln. Bei LOW ist das ROM aktiv und liefert Daten. Pinout der ROM's siehe unter Chips.

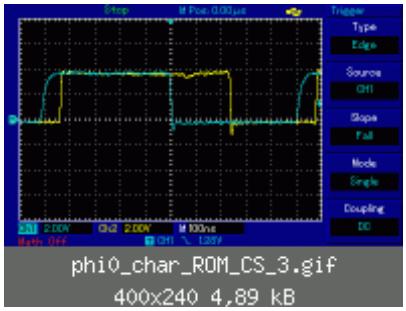
Messung von _CS an den ROMs: Im Leerlauf wird das BASIC-ROM (901226-01) nicht angesprochen, man kann aber einfach über das Drücken der RETURN-Taste Zugriffe produzieren.

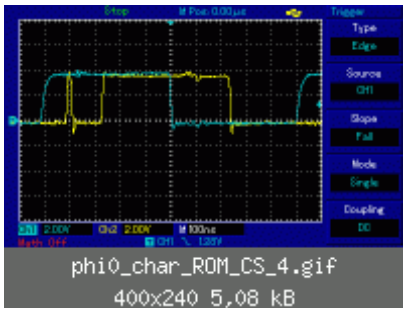
Oszi auf 'single' einstellen und auf HIGH -> LOW triggern lassen. Man sollte bei jedem Start der Messung sofort eine Flanke einfangen können.

Hier noch 4 Bilder. Sie stellen das _CS Signal des Char-ROM (Pin 20) relativ zu Phi0 (CPU Pin 1) dar. Der Rechner funktioniert einwandfrei und die PLA ist ein Original von MOS. Die Bilder lassen vermuten, dass die Logik innerhalb der Original-PLA auch nicht wirklich 100% sauber ist. Manche Merkwürdigkeiten scheinen aber nicht zu stören:







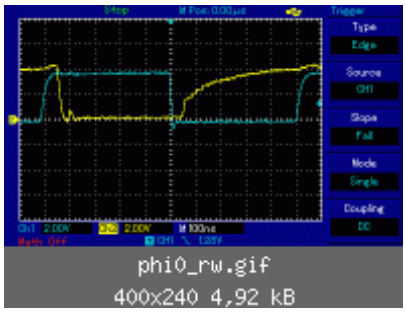


Kommen wir zum R/_W-Signal (CPU Pin 38) . Damit zeigt die CPU an, daß sie Daten schreiben will. Der VIC kann nur lesen, bei ihm dient R/_W als reiner Eingang für den Zugriff auf seine Register.

Für das Bild brauchte ich wieder die Einstellung 'single' und ein paar Versuche bis ich einen Schreibzyklus erwischt habe.

Wie man sehen kann steigt R/_W am Ende des Zyklus erst schnell an und dann deutlich langsamer. Der Knick kommt vom AEC-Signal welches die CPU vom Bus nimmt wenn der VIC dran ist. Ab dort wird das Signal nur noch vom Pullupwiderstand der R/_W-Leitung (1.5KOhm) langsam hochgezogen. Das ist nicht wirklich optimal weil die R/_W-Leitung direkt an den RAMs hängt und bei zu starker Belastung der Leitung der Anstieg zu langsam sein könnte und damit die RAMs den VIC-Zyklus als Schreibzyklus sehen könnten. Es funktioniert offensichtlich, aber schon beim 250466 hat Commodore mit U11 und J3 einen

Workaround vorgesehen (aber nicht implementiert) und beim 250469 das R/_W-Signal für die RAMs durch die PLA geschickt:

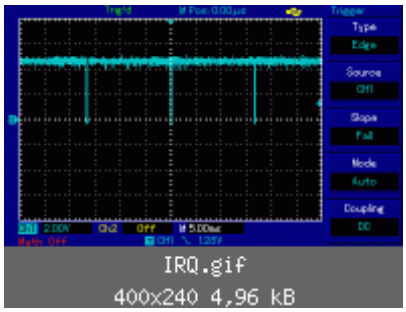


Jetzt der Datenbus (Pin 37/D0 - 30/D7 an der CPU)

Es sollte Aktivität zu sehen sein und zwar auf allen 8 Leitungen. Eine oder mehrere Leitungen die sich nie ändern deuten auf ein Problem mit irgendeinem Chip der mit dieser Leitung verbunden ist hin. z.B. ein defektes RAM, CIA, SID, VIC oder ROM welches eine Leitung dauerhaft nach LOW zieht. Meist (nicht immer) erkennt man solche Defekte an großer Hitzeentwicklung des betroffenen ICs.

Hier ein Bild von D0 in Relation zu PHI0. Der kleine Überschwinger auf D0 kurz nachdem PHI0 nach LOW wechselt war bei meinem Board problemlos reproduzierbar, beeinflusste den Betrieb aber nicht.
Zum _IRQ-Signal. Damit wird der CPU angezeigt, daß ein andere Gerät etwas will. Das Signal ist inaktiv solange es HIGH ist. Bei einem C64 passieren schon im Leerlauf (nur eingeschaltet) 60 IRQs pro Sekunde vom Systemtimer über den auch die Software-Uhr (TI und TI$) geführt werden. Dieser Timer ist in CIA1 zu finden an dem auch die Tastatur hängt. Auch das Blinken des Cursors wird damit über Software realisiert.

Man sollte also bei eingeschaltetem C64 an Pin 3 der CPU Aktivität feststellen können. Die LOW-Zeiten sind in Relation ziemlich kurz, also die horizontale Ablenkung auf 5ms pro Feld einstellen, damit kann man 3 IRQs sehen:

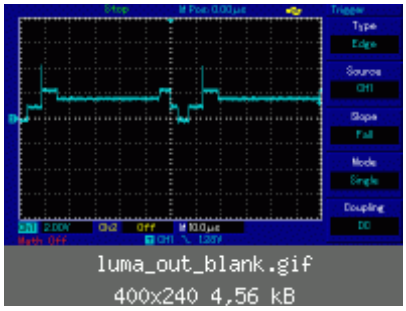


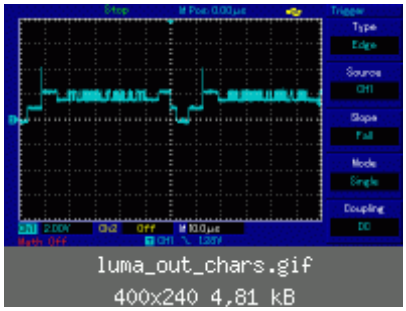
Der Ausgang des VIC. Der Video-Ausgang des VIC wird durch den Modulator geroutet und dort im Pegel noch angepasst. Sollte dort ein Schaden sein sieht man auch am Videoausgang nichts obwohl das System sonst OK ist.

Hier zwei Bilder, aufgenommen am VIC Pin 15 (Sync/Luma) mit 10µs pro Feld. Einmal mit einem Teil der Einschaltmeldung und einmal ohne. Jeweils aufgenommen mit 'single'. Man kann die Synchronsignale recht gut erkennen.

Fehlendes Chroma (Pin 14 am VIC) ergibt ein Graustufenbild bei sonst einwandfreier Funktion.

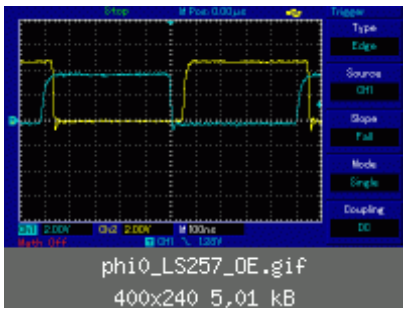
Der sehr kurze, hohe Peak am Anfang der dargestellten Zeile müsste die weisse senkrechte Linie am linken Bildrand sein:





Hier noch ein wichtiges Signal. Damit der VIC ungestört seine Zugriffe machen kann müssen die Multiplexer (2 x 74LS257) ihre Ausgänge abschalten, Dieses wird bei den älteren Boards über das AEC-Signal und ein Gatter des 7406 mit einen sehr kleinen Pullupwiderstand (180Ohm) gesteuert, Diese Schaltung ergibt ein invertiertes AEC-Signal. Das Freigabesignal ist an den 74LS257 an Pin 15 zu finden und muss LOW sein wenn die CPU auf das RAM zugreifen darf. Ist der 7406 defekt und zieht das Signal dauerhaft auf LOW kämpfen die 74LS257 und der VIC um die Hoheit auf dem DRAM-Adressbus was garantiert Müll auf dem Monitor ergibt. Ist das Signal dauerhaft HIGH bekommt die CPU gar keinen Zugriff auf das DRAM und der Rechner startet nicht.

Das Bild zeigt PHI0 in blau und _OE (Pin 15) der Multiplexer in gelb. Wie die meisten Bilder in 'single' während einer normalen Bildschirmzeile mit abwechselnden CPU- und VIC-Zugriffen aufgenommen. Bei einer Badline wäre es für die Dauer der Badline durchgehend HIGH.



Hier noch ein Hinweis zu den Messungen.

Ich benutze ungerne den Tastkopf am Oszi wenn ich es vermeiden kann, man rutscht zu schnell ab und verursacht Kurzschlüsse womit man den Rechner noch weiter beschädigen kann. Der Klemmaufsatz ist aber etwas zu grob für DIP, der taugt bei meinem Oszi eigentlich nur für die Pins an den Ecken der ICs.

Also habe ich mir aus Pollin Best. Nr. 830 262 und einem Stück Litze einen kleinen Adapter gebastelt. Damit kann man sauber messen, das Signal wird bei den im C64 üblichen Frequenzen gegenüber der direkten Anwendung das Tastkopfes nicht sichtbar verzerrt. Die Klemme nur bei ausgeschaltetem C64 an das zu messende Signal anklemmen, verifizieren das kein Kurzschluss entsteht und dann erst einschalten und auf den Bildschirm des Oszi schauen.

GND für die Messungen hole ich mir meist vom Modulator, da findet sich immer ein Platz den dem die Krokoklemme hält.

Oszi und Tastkopf sind bei mir beide auf 'x10' eingestellt.

Das Bild zeigt die Messung von _IRQ an meinem Testboard. Wurde beim Vorbesitzer über Jahre offen im Keller gelagert. Funktioniert trotz des Aussehens aber einwandfrei nachdem ich einige defekte ICs gewechselt habe.

C64 Ersatz-PLA auf 27512 Eprom Basis

Nach Joachim Nemetz (Jogi) und Jens Schönfeld

Der hier beschriebene Adapter basiert auf der Arbeit von Jens Schönfeld, dem Geschäftsführer der Individual Computers GmbH. Im August 1994 hat er als Zwischenschritt zur Logikanalyse des PLA-Chips einen Adapter gebaut, der das Auslesen mit einem Eprom-Programmiergerät ermöglichte. Die entstandene Datei wurde im Z-Netz (einem Vorläufer der Internet-Foren) veröffentlicht. Es gab zahlreiche Anläufe zur Analyse von anderen Leuten, jedoch war es wieder Jens Schönfeld, der im Sommer 2002 die Logik-Gleichungen der C64-PLA veröffentlichte. Sie sind im Funet-Archiv zu finden, das sowohl kostenlos online, als auch als CD im Handel erhältlich ist. Dort sind auch weitere Eprom-Dateien zu finden, die mit diesem Adapter angefertigt wurden, denn Commodore hat den Logik-Chip 82S100 in vielen Computertypen eingesetzt - immer mit unterschiedlicher Programmierung.

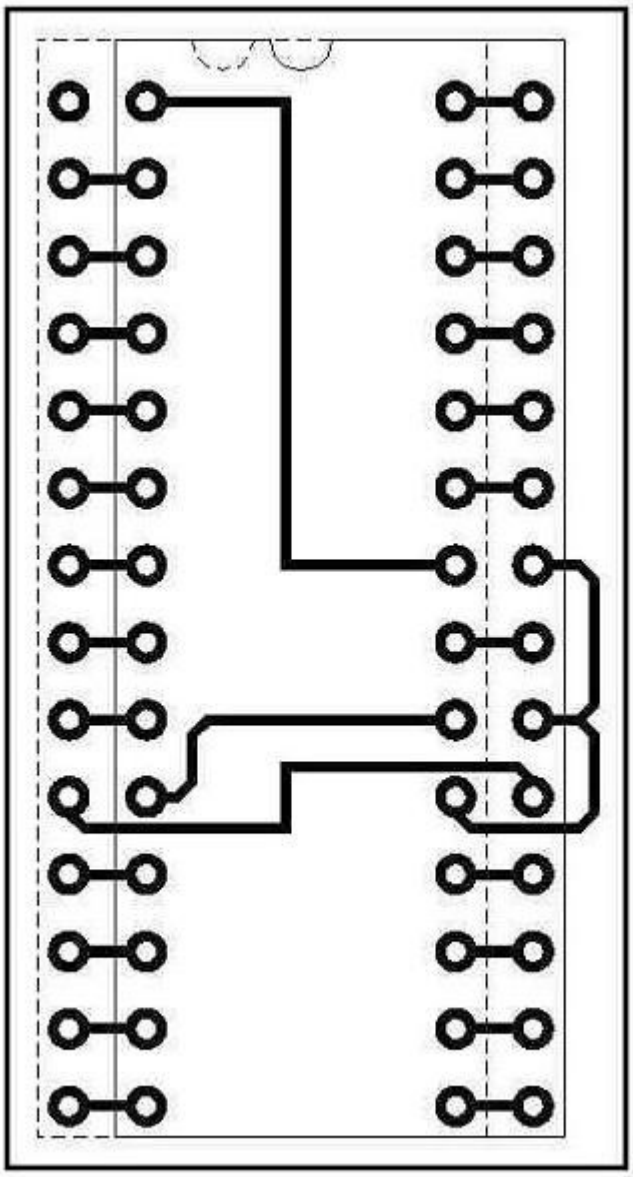
Der vorgestellte Adapter nutzt die Datei, die einmal als Zwischenschritt gedacht war nun als Quelle für eine Ersatzschaltung. Dabei wird ausser Acht gelassen, dass ein Eprom als Logik-Ersatz eigentlich nicht geeignet ist. Während der Zugriffszeit von typischerweise bis zu 120ns (also gut einem Viertel des CPU-Zyklus des C64) ist der Zustand der acht Ausgangsleitungen nicht garantiert. Unter Umständen kann es vorkommen, dass zwei oder gar noch mehr Bausteine des C64 die Freigabe bekommen, den Datenbus des C64 zu benutzen. In den meisten Fällen geht das gut, jedoch sind Langzeitschäden nicht auszuschließen. Die Eprom-Schaltung sollte demnach nur als Übergang oder zur Diagnose eingesetzt werden. Für eine dauerhafte Lösung ist dringend der Bau einer Ersatzschaltung auf Basis eines programmierbaren Logik-Chips angeraten.

Wenn der C64 (Ur-Modell Bj. 82-84) den Geist aufgibt ist zu 90% die PLA platt. Der Baustein hat die Bezeichnung 906114, bzw. 82S100. Abhilfe schafft die "Bastel-PLA" die aus einer Adapterplatine und einem 64K EPROM besteht.

Benötigt wird dazu ein Eprom 27C512-45ns wobei bei mir 120ns, 150ns, 200ns und 255ns ohne sichtbare Probleme liefen im C64 - also einfach Testen, eine Adapterplatine oder ein freiverlöteter Adaptersockel und natürlich die passenden Daten

-> http://petersieg.webng.com/pla/27512.zip

Bild der Verbindungen:

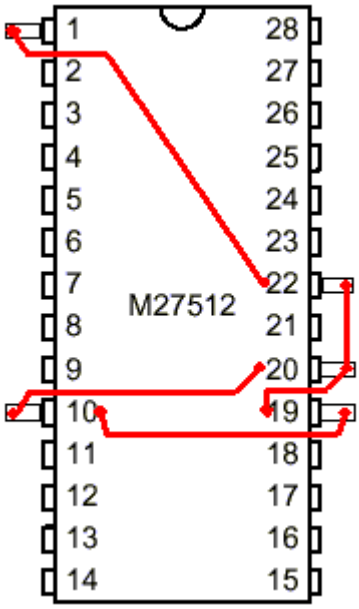


Das sollte sich auch 'fliegend' mit Zwischensockel aufbauen lassen:

Dazu folgende Erklärungen

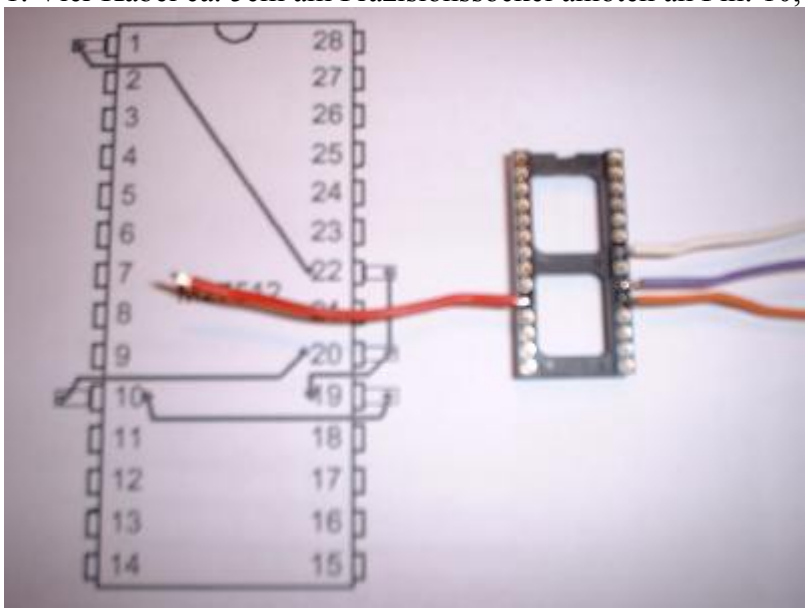
In den Sockel des C64 Boards kommt ein Zwischensockel (normaler IC DIL28 Sockel). An den PIN's: 10, 19, 20, 22 wird ein isoliertes Kabel/Litze angelötet, sodaß man das andere Ende dann später an die abgebogenen Kontakte des 27512 Eproms löten kann. Dann kommt das programmierte! 27512 Eprom auf den Zwischensockel, wobei am Eprom die Pin's: 1, 10, 19, 20, 22 zur Seite gebogen werden, sodaß sie keinen Kontakt mit dem Zwischensockel haben! Dann werden verbunden:
Zwischensockel <-> Eprom

10 <-> 19
19 <-> 20+22
20 <-> 10
22 <-> 1


1
2
3
4
5
6
7
8
9
10
11
12
13
14
28
27
26
25
24
23
22
21
20
19
18
17
16
15
M27512

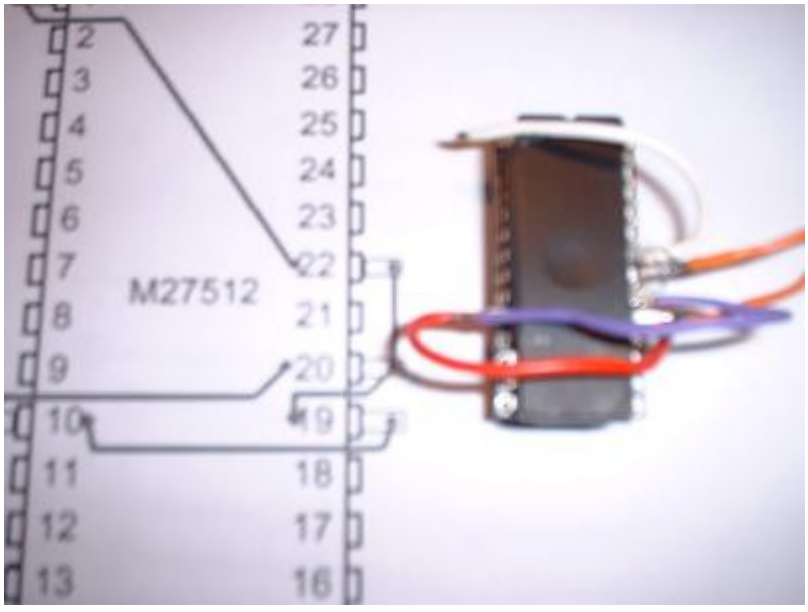
Hier ein Aufbau

1. Vier Kabel ca. 5cm am Präzisionssockel anlöten an Pin: 10, 19, 20, 22.



2. Dann ein 27C512 mit den korrekten Daten brennen.
Dann PIN 1, 10, 19, 20, 22 abbiegen, sodaß sie keinen Kontakt zum Sockel haben!

3. Die 4 Kabel wie oben ausgeführt mit dem Eprom-Pins verbinden:
Sockel -> Eprom
10 -> 9
19 -> 20+22
20 -> 10
22 -> 1

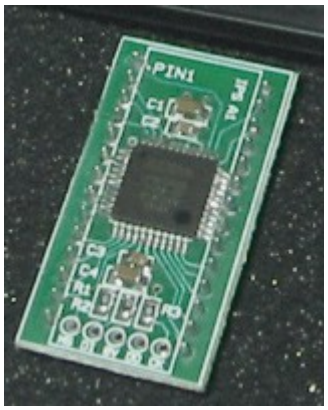


Ersatz-PLA auf CPLD Basis

Link: http://www.fpgaarcade.com/c64_customs.html

Basierend auf den Arbeiten von Jens Schönfeld und Mark Smith hat Armin Läuger die Logikgleichungen für die C64 PLA auf das CPLD Board von Fpgaarcade portiert.

Das CPLD Board ist im DIL28 Formfaktor und kann zum Ersatz verschiedenster IC's programmiert werden.



Referenzen:

[1] http://www.zimmers.net/anonftp/pub/cbm/firmware/computers/c64/
[2] http://vic20.de/html/eprom_pla_8296_und_c64.html
[3] http://www.zimmers.net/cbmpics/cbm/c64/pla.txt

C64 Restaurationen

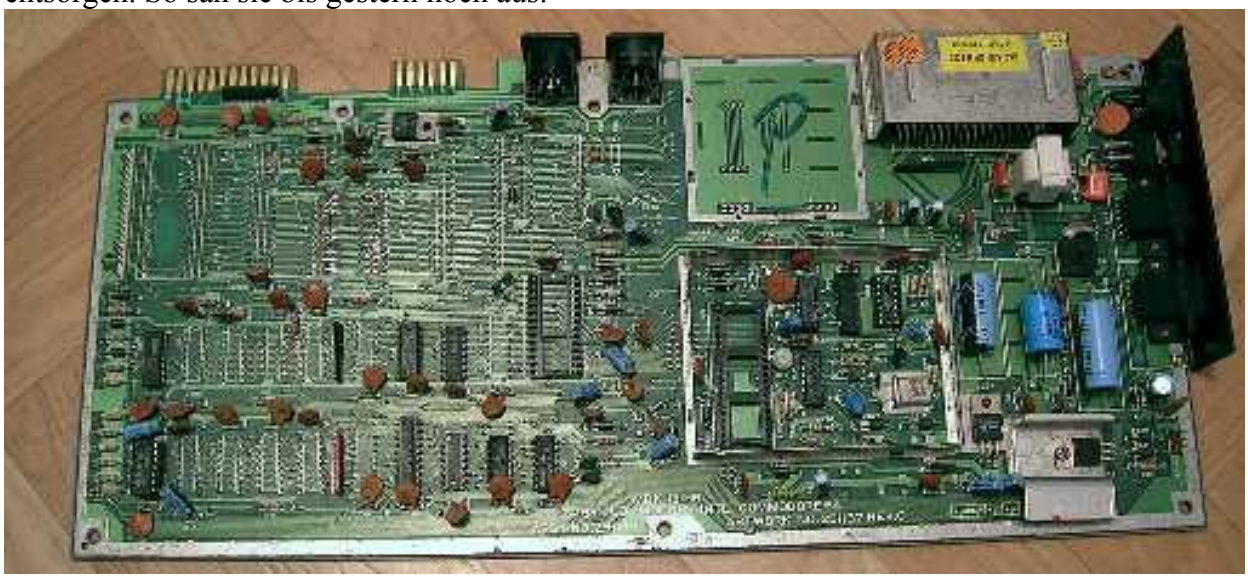
C64 Assy 250407 Wiederbelebung

Diese großartige Wiederbelebung wurde ausgeführt und hier mit freundlicher Genehmigung wiedergegeben von Andreas Senk (JerryMaus).

Hallo,

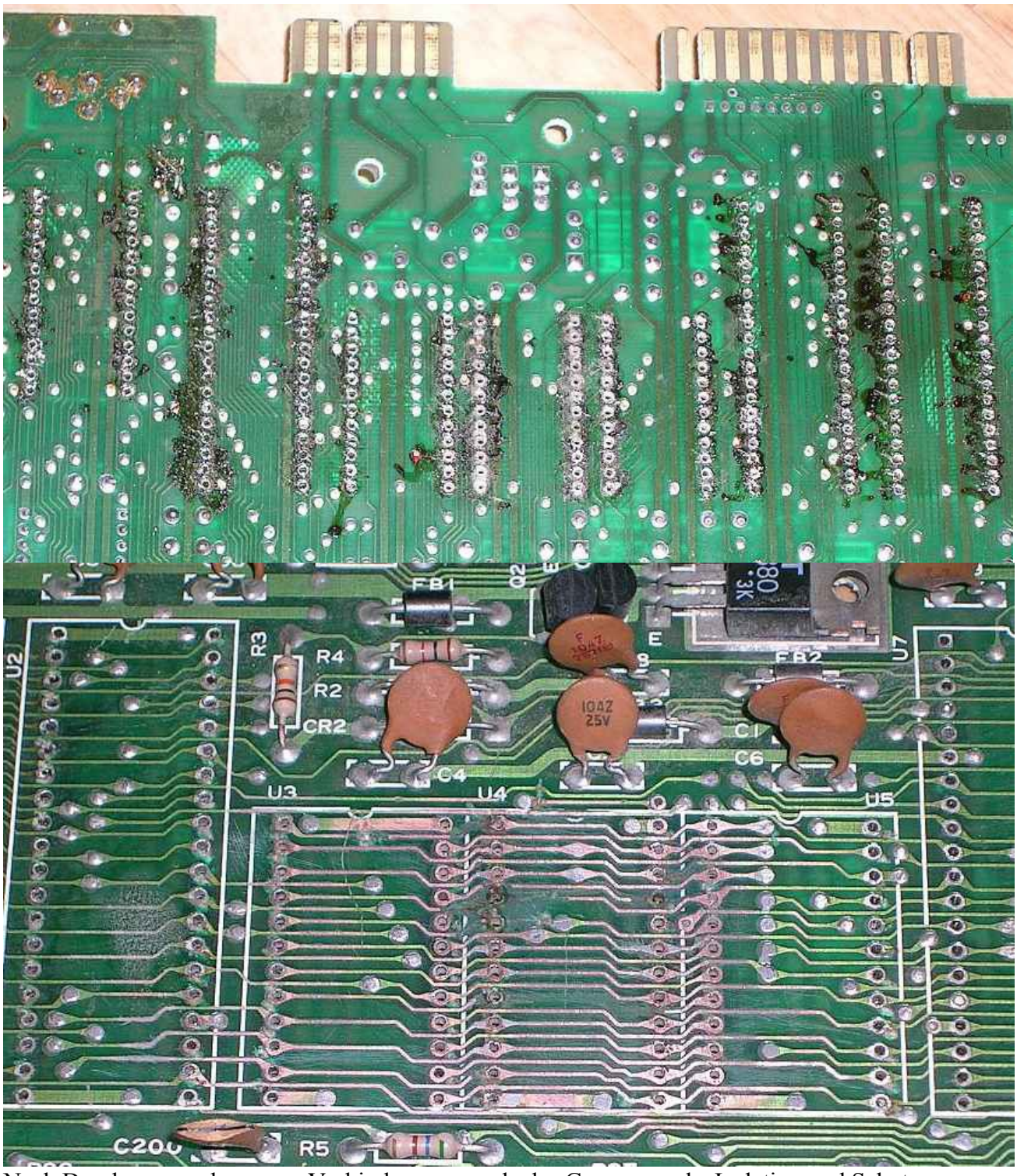
ich habe diese Platine Ende der 80´er bekommen. Der vorherige Eigentümer hat versucht, das Kernal auszulöten und dabei mehrere Durchkontaktierungen und Leiterbahnen abgerissen. Das Gehäuse, Tastatur und viele der Bauteile sind über die Jahre anderen 64´ern zugute gekommen.

Da nun so ziemlich alles brauchbare nicht mehr vorhanden war, wollte ich den Rest eigentlich entsorgen. So sah sie bis gestern noch aus:

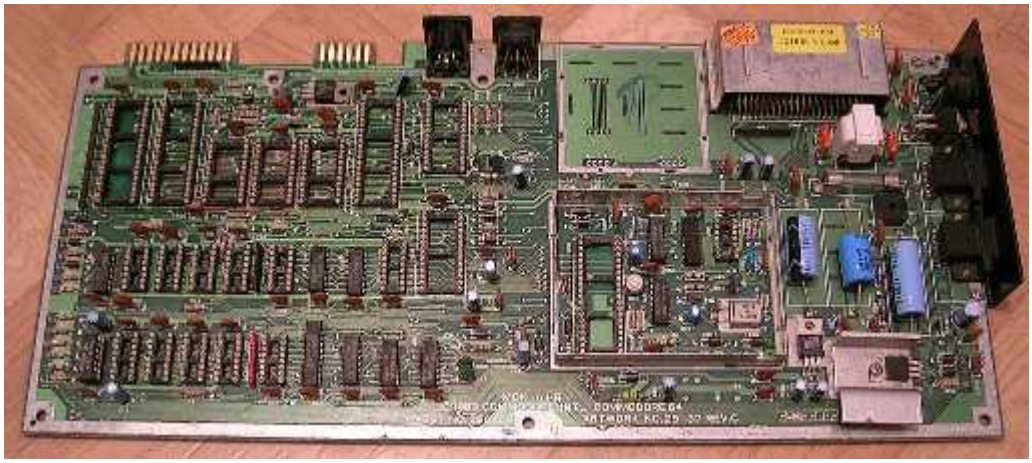


Inspiriert durch Pentagon´s Bericht über die Rettung eines C64G habe ich es dann doch nicht getan, sondern mich zu einem Rettungsversuch entschlossen.

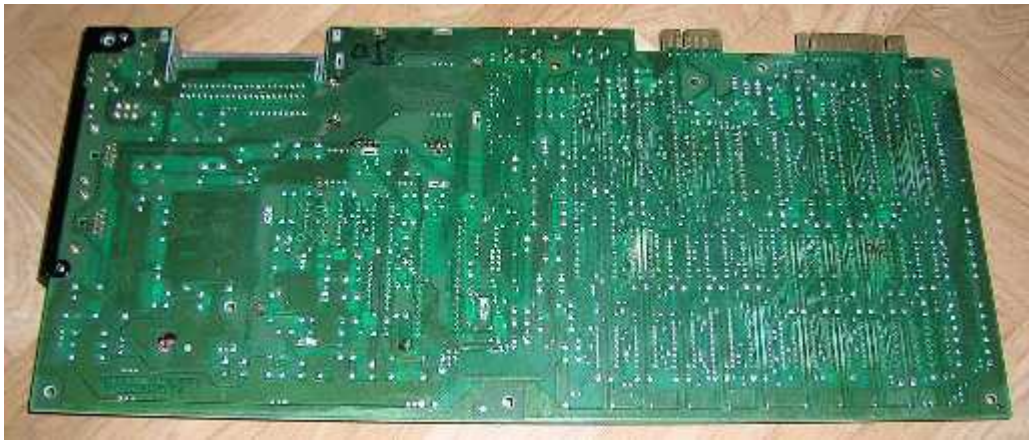
Nach Reinigen der Platine und Abschleifen der defekten Leiterbahnen mit einem Glasfaserpinsel war der Schaden gut zu sehen:



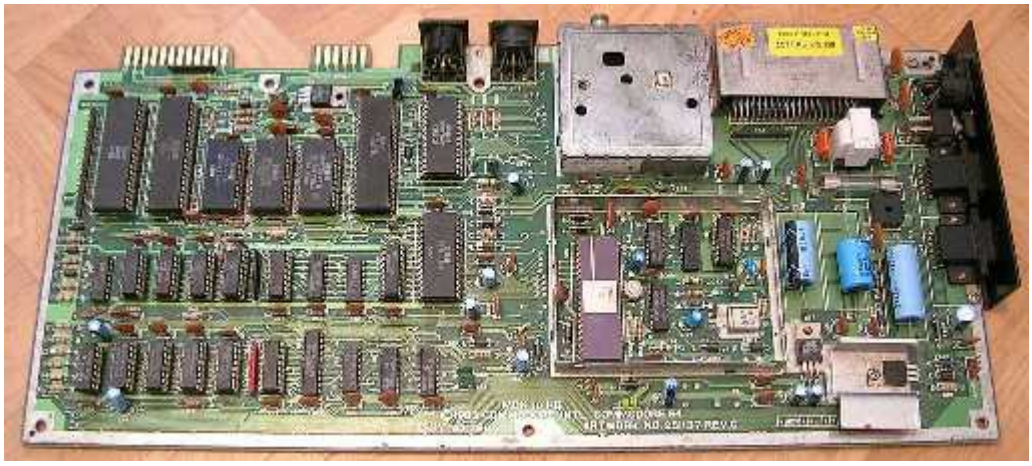
Nach Durchmessen der neuen Verbindungen wurde das Ganze zwecks Isolation und Schutz vor Korrosion mit einem durchsichtigen Schutzlack überzogen. Heute war dieser trocken und ich habe Sockel eingelötet. Im Zuge dessen wurden auch gleich die alten, noch vorhandenen Sockel durch Präzisionsfassungen ersetzt:



Und so schaut nun die Rückseite nach dem Reinigen, Löten und Versiegeln aus:



Die Bauteile habe ich aus meinem Fundus zusammengesucht, eine CIA und den SID jedoch vorerst einem anderen Rechner entliehen:



Der anschließende Funktionstest fiel nicht so gut aus. Es hatte sich noch eine Unterbrechung bei den Speicherbausteinen eingeschlichen. Diese war aber schnell gefunden. Die Bauteile hatte ich natürlich vorher in einem anderen Cevie einzeln auf Funktion getestet. Und siehe da:

Ein brauchbares Gehäuse sowie eine Tastatur fehlt leider noch. Aber die Platine hat schon mal ein neues Leben vor sich.

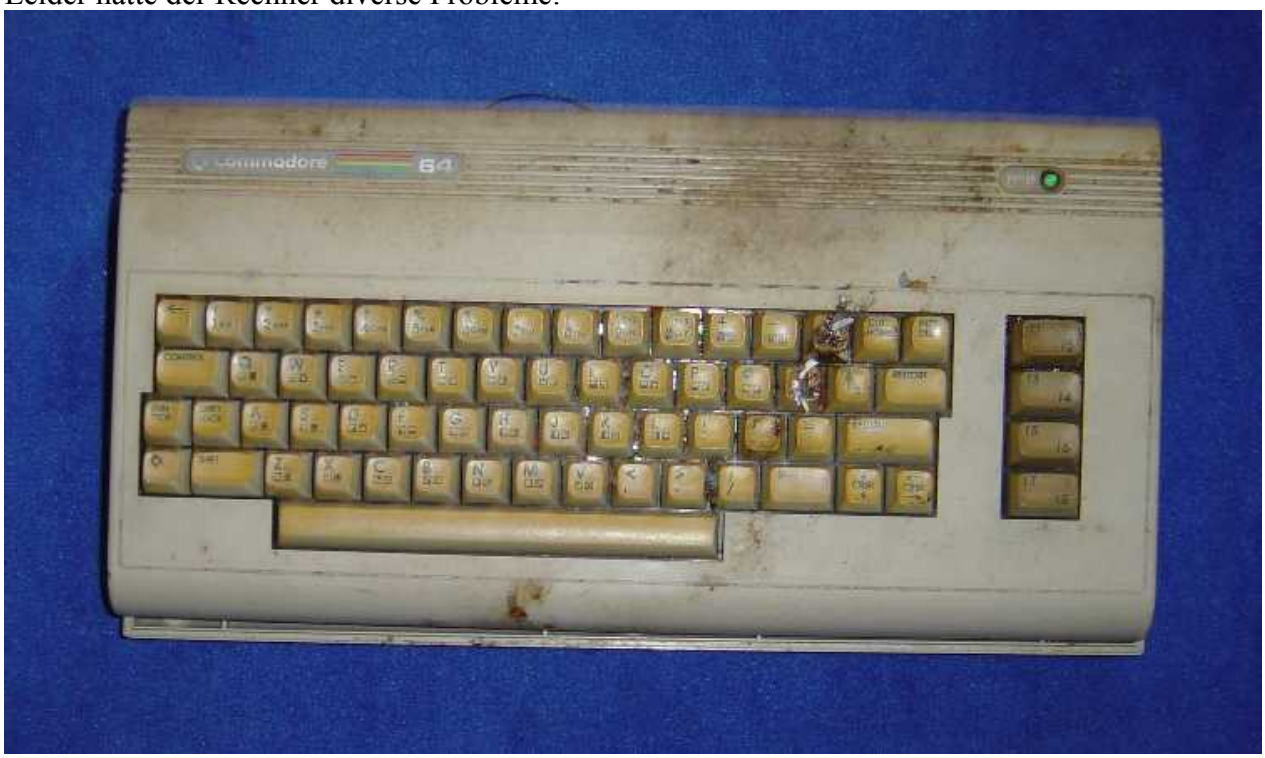
Andreas Senk

C64 Restauration Pentagon

Diese großartige Restauration wurde ausgeführt von Thomas Wirtzmann (Pentagon).

Und hier kommt die Story: Nach langer Zeit hat mich unser geschätzter Doc64 letzte Woche mal wieder zu Hause besucht. Wie es sich gehört hat er mir als freundliches Gastgeschenk - einen Commodore 64 mitgebracht. Es war ein wunderschöner C-64G aus dem Jahre 1990. Und die Freude meinerseits war natürlich riesengroß. Na ja nicht ganz, wie man sich denken kann, rückt der Doc nicht so ohne weiteres einen Cevie raus, oder?

Leider hatte der Rechner diverse Probleme:



Manfred war am Morgen zu einem Bekannten gefahren und hatte das Gerät dort im Keller gefunden, wo es zwischen Müllbergen, Mäusekot, Schimmel und Dreck sein trauriges Endzeitstadium fristete. Da ihm das gute Stück extrem leid tat, hat er es kurzerhand eingepackt und beschlossen mir den Rechner als Geschenk mitzubringen, da ich immer auf der Suche nach Ersatzteilen bin.

Als ich das Gerät dann in die Hände gedrückt bekam, war mein Entsetzen groß, sowaß hatte ich nämlich noch nie im Leben gesehen. Abgesehen von dem bestialischen Gestank von Moder und Kleintierexkrementen, war das Teil einfach nur ekelhaft. Mit spitzen Fingern haben wir dann das Gerät aufgeschraubt und wollten die Platine zum späteren schlachten ausbauen. Nachdem öffnen des Rechners kam der nächste Schock, die Platine hatte einen Schaden durch eingedrungene Flüssigkeit, vermutlich Coca Cola. Außerdem war noch eine Art stinkender klebriger Leim rund um den Modulport, welche so ekelhaft gerochen hat. Es würde mich nicht wundern, wenn es eine Verwesung war.

Wir haben die Platine trotzdem ausgebaut und das ekelhafte Gehäuse in meine Restmülltonne befördert. Denn es war einfach nur noch schrottreifer Müll.

Die Platine wurde sofort bei uns ins Badezimmer gebracht, wo sie sogleich in 70 Grad heißes Wasser getaucht wurde. Es war ein Jammer, aber der Geruch war unter Wasser nicht mehr

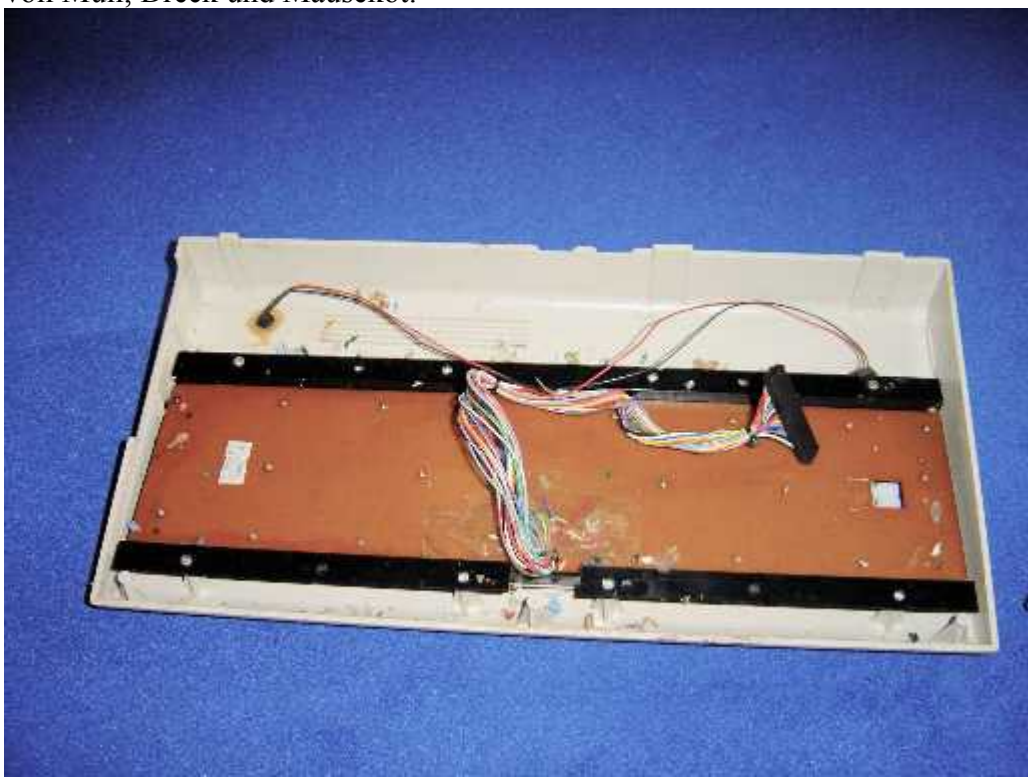
wahrzunehmen. Dann haben wir ein Schaumbad aus Spüli und Duschgel zubereitet und die Platine nach allen Regeln der Kunst mit Zahnbürste und Nagelbürste geputzt. Nach knapp 40 Minuten dauerschrubben, war die Platine dann soweit sauber und mir war zum schwallartigen Erbrechen zumute. Sowas ekelhaftes, habe ich noch geputzt. Das Blech vom Modulport war so verrostet und angefressen, dass es grosse Blasen geworfen hatte. Die Drosselspule war weggefressen und der Schalter ließ sich überhaupt nicht mehr richtig bewegen und stank selbst nach dem Bad noch ungemein.

Die Platine habe ich dann in einem Zimmer auf die Heizung gelegt und wir haben uns anschließend einen schönen Abend gemacht und 80er Jahre Videos geschaut. Als Manfred dann später nach Hause fuhr, habe ich mir die trockene Platine angesehen und festgestellt, daß diese erst 17 jahre alt war, also relativ jung für einen C-64. Es waren nur CSG Chips aus dem Jahr 1990 verbaut und das Herstellungsdatum war Kalenderwoche 40 / 1990.

Ich bin dann ziemlich müde ins Bett gekrabbelt und hatte eine sehr unruhige Nacht. Irgendwie hatte ich ein schlechtes Gewissen bekommen, dass ich das C-64 Gehäuse so einfach in den Müll geworfen hatte. Meine Frau hat mich für total bescheuert erklärt und sagte, dass dieses Ekelteil auf jeden Fall in der Mülltonne bleibt und entsorgt wird.

Früh morgens, war es dann soweit. Meine Unruhe war so groß, ich bin dann mit meiner Schlafanzughose und einem Pulli an die Mülltonne geschlichen. Dann öffnete ich den Deckel der stinkenden Tonne und nahm das widerliche Gehäuse wieder raus. Im gleichen Moment kam meine Nachbarin um die Ecke und hat mich mit dem Ding an der Tonne stehen sehen. Nachdem sie dann noch meine Schlafanzughose wahrnahm, hat sie den Kopf geschüttelt und ist fassungslos wieder zurück ins Haus gegangen. Ihre Gedanken konnte man förmlich am Gesichtsausdruck ablesen.

Nachdem ich das gute Stück dann mit ins Haus genommen hatte, habe ich es erst einmal in die Badewanne verfrachtet und auseinander genommen. Im oberen Gehäusedeckel war alles voll Müll, Dreck und Mäusekot:



Die Tastatur war auch total verklebt und ein schleimiger Brei klebte rund um die F-Tasten und über der Return Taste. Das ganz war so verharzt, dass man die Tasten nicht mehr

bewegen konnte:

Nach dem Ausbau der Tastatur habe ich dann das Gehäuse in der Badewanne mit der chemischen Keule gereinigt. Ich habe einen Spezialschaum eingesetzt, welcher auch in die Poren einzieht und gleichzeitig leicht bleichend wirkt. (Anmerkung: Wasserpfeifenreiniger "Schmand Weg" in lauwarmen Wasser soll hier gute Ergebnisse bringen)

Nachdem Einsatz der chemischen Keule, wurde das Gehäuse dann geschrubbt und das erste Ergebnis war schon mal gar nicht schlecht. Wenn auch noch lange nicht perfekt.

Nach fast zweistündigem schrubben, putzen, bleichen, trocknen, polieren war ich mit dem Ergebnis soweit zufrieden.

Jetzt habe ich mich der Platine gewidmet. Die Platine war defekt und funktionierte überhaupt nicht mehr. Also habe ich den Schalter abgelötet, die Spule entfernt und das Modulblech ebenfalls entfernt. Den verfaulten Sicherungshalter, habe ich ausgelötet und ersetzt.

Der Schalter wurde aufgebogen und zerlegt. Im inneren befand sich noch Wasser und klebriger Brei, welcher so zäh war, dass man ihn nicht entfernen konnte. Also habe ich den Schalter ausgetauscht und einen neuen eingelötet. Die Platine war soweit wieder sauber und sah aus wie gestern produziert. Die Spule habe ich neu eingelötet und die Sicherung ersetzt. Die gesockelten Chips habe ich gereinigt und neu eingesetzt.

Jetzt kam der große Augenblick und ich habe die Hauptplatine dann am 1701 angeschlossen. Eingeschaltet und:

Sie funktionierte wieder.

Ein Modulportblech, war auch noch in meinem Ersatzteilbestand, zwar vom C-64E aber egal. Hauptsache es passt.

Sieht doch wieder sehr gut aus und funktioniert auch wieder einwandfrei. Der Modulport war mit Grünspan oxidiert und wurde mit dem Dremel und einem Zahnarztpolierer wieder vorsichtig auf Hochglanz gebracht. Steckmodule funktionierten auch wieder einwandfrei.

Jetzt konnte die fertige Hauptplatine wieder in die Unterschale eingebaut werden. Die Unterschale hatte auch einen Defekt. Beim Ausbau der Platine ist oben rechts eine Gewindeführung abgebrochen. Diese hing noch am Metallblech von den Ports dran. Also habe ich sie mit Sekundenkleber wieder eingeklebt und die Naht mit einer kleinen Lötkolbenspitze verschweißt. Das hält auf jeden Fall wieder sehr gut und die Platine konnte einwandfrei verschraubt werden. Selbstverständlich habe ich die verrosteten Schrauben gegen neue ersetzt.

Die Silberpappe für die Abschirmung, war leider total verschimmelt und konnte nicht weiter verwendet werden. Also verzichte ich eben darauf. Wird auch ohne Pappe funktionieren.

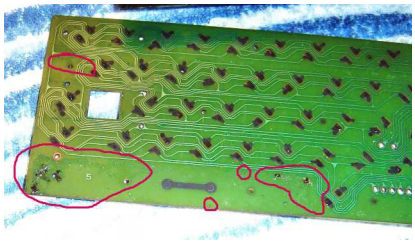
Jetzt habe ich die Tastatur in der Badewanne mit dem Schaumreiniger eingesprüht und habe das einwirken lassen.

Das war wirklich extrem ekelhaft, beim Abspülen offenbarte sich der gesamte Dreck der letzten 17 Jahre. Dort waren tote Insekten, Würmer, Dreck, Mäusekot und Abfallteile zwischen den Tasten. Es wurde immer mehr Dreck. Unglaublich wieviel Dreck in so einer Tastatur vorhanden sein kann. Das war sowas von krass, dafür gibts keine Worte.

Nach dem ersten Abspülen, habe ich dann die Tastenkappen entfernt und die Tastatur komplett zerlegt. Dabei hätte ich mich wirklich fast übergeben müssen, denn im inneren der Tastatur befanden sich kleine Würmer und winzige Käfer die sich dort einen Lebensraum

erschaffen hatten. Ich hätte das nie für möglich gehalten, aber diese Insekten haben dort auf der Platine lauter Eier abgelegt und es lebte.

Hier kommt der widerliche Beweis:



Die Tastenkappen habe ich dann in Oxycleanbad gegeben und über Nacht stehen lassen. Das Oxyclean, löst den Dreck auf und bleicht die Tasten. Leider war die Bleichwirkung so gering, daß ich zu härteren chemischen Mitteln greifen mußte. Ich habe reine Chlortabletten mit kochendem Wasser aufgegossen und eine 70% Chlorlösung gebraut. Das ganze noch mit einer Flasche Domestos aufgegossen und fertig war die Chlorgasbombe.

Das ganze habe ich natürlich draußen gemacht, denn der Gestank ist nicht nur unerträglich, sondern auch hochgiftig. Also bitte nicht zuhause nachmachen. Mein Nachbar war ganz verwundert, über den penetranten Schwimmbadgeruch. Ich habe mir nichts anmerken lassen und innerlich gelacht. (Anmerkung: Vorsicht mit solchen chemischen Substanzen! Sicherheitshinweise beachten!)

Die Tasten haben es überlebt und sind schön gebleicht. Leider hat mir das Ergebnis trotzdem nicht gefallen. Also habe ich mich entschlossen, einen Satz neue Tastenkappen zu verwenden.

Ist schon ekelig oder?

Nach der Reinigung und Aufbereitung der Tastatur konnte diese wieder in das Gehäuseoberteil eingebaut werden. Die LED des Cevie war auch hinüber. Sie war so verrostet, das der Rost sich bis ins innere gefressen hatte. Da ich keine grünen LED's im Haus hatte, musste ich notgedrungen eine blaue verwenden. Ich hoffe man verzeiht mir das

"Modding". Die Kabel zur LED wurden ebenfalls ausgetauscht, da sie ebenfalls brüchig vor Korrosion waren. Neuer Schrumpfschlauch wurde ebenfalls "drumgeschrumpft".

Nach der Montage des Gerätes sah es dann so aus:

Das Gerät war nun nach langer Arbeit endlich fertiggestellt. Nicht ohne Stolz habe ich es dann sofort angeschlossen und habe es jetzt seit vier Stunden im Dauerbetrieb. Als meine Frau dann von der Arbeit nach Hause kam, wurde das Gerät sofort besetzt und mit großer Freude begrüsst.

Alles in allem ist es doch sehr schön geworden. Jemand anderes, hätte das Ding mit Sicherheit im Müll gelassen. Aber wie man sieht, lohnt sich Ausdauer und Fleiß und man bekommt ein gutes Ergebnis.

Vielleicht regt es Euch an, eventuell haltet Ihr mich für bescheuert. Der Cevie funktioniert auf jeden Fall wieder.

HAPPY BIRTHDAY !!!

Pentagon

Großumbau C64 (Verringerung der Stromaufnahme; Lebensverlängernde Maßnahmen)

Diese großartige Umbau wurde vorgenommen von Peter Trachte (AREA51HT).

Generell sollte man auch ein neues Netzteil am C64 einsetzen. Da hier aber mit Netzspannungen hantiert werden muß, sollte dies nur von einem Fachmann ausgeführt werden! Belegung der Netzteilbuchse/Kabel ist aus der kleinen Reparaturanleitung zu entnehmen. Es werden +5V ca. 1.5A und ~9V va. 1A benötigt.

Der Umbau ist nicht für das seltene Sammlerstück gedacht, sondern für die Arbeitsmaschiene die viel läuft.

Ziel der vielen kleinen Baustellen ist es, die Verlustleistung im Innern des Gehäuses zu reduzieren um die Betriebstemperatur der Chips abzusenken um die Lebensdauer zu maximieren.

Was bringt der Umbau:
Stromaufnahme vor dem Umbau:

Original
9V 0,65A =5,85 Watt
5V 0,85A =4,25 Watt

_____
10,10 Watt Gesamtleistung

Nach dem Umbau
9 V 0,326A =2,94 Watt
5 V 0,40A =2,00 Watt

_____
4,94 Watt Gesamtleistung

Die Messergebnisse:

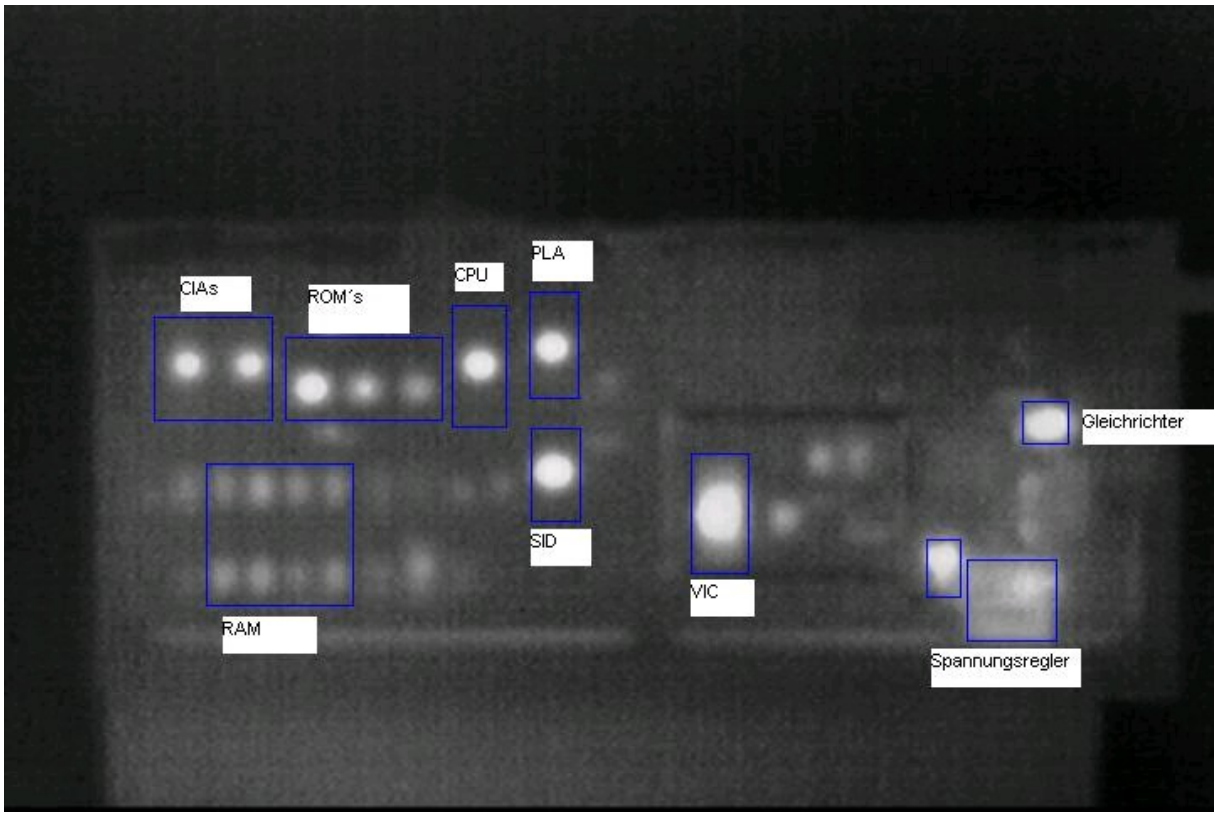
Meßaufbau	Luft	Core	Delta	Oberfläche	Innenluft
1 Brotkasten mit Schirm	23,0	93,3	70,3	82	42
2 Brotkasten ohne Schirm	22,0	85,3	63,3	76	39
3 Brotkasten ohne Schirm mit losen Kühlkörper	21,5	72,5	51,0	63	38
4 Brotkasten ohne schirm mit wärmeleitpast	22,0	71,2	49,2	62	38
5 Brotkasten Ohne Schirm + Kühlkörper mit Low Power Umbau	21	72	45	--	27

Core = Silizium Die Temperatur
Delta = Temperatur Differnz zwischen Umgebungsluft aussen und Silizium Die
Luft = Lufttemperatur ausserhalb C64 Gehäuse
Oberfläche = Oberflächentemperatur SID oder Kühlkörper
Innenluft = Lufttemperatur innerhalb des Gehäuses

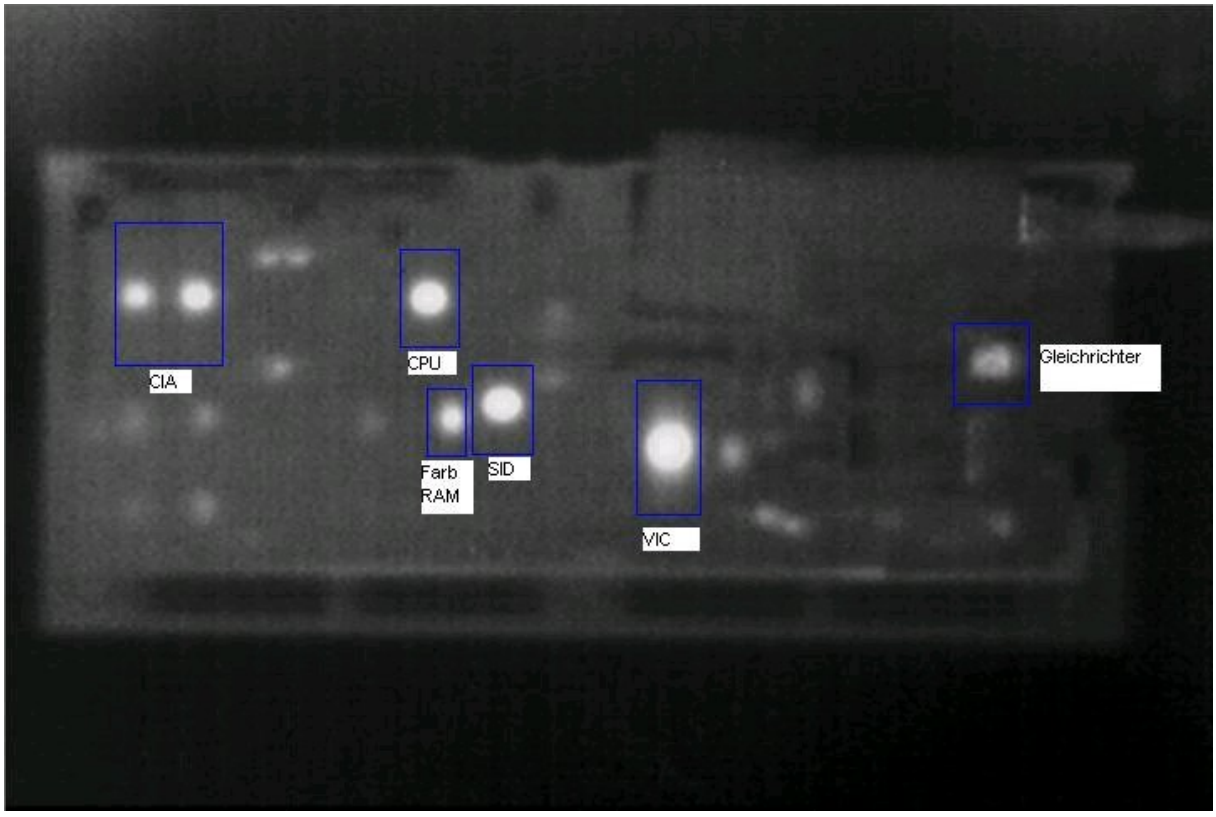
Wir sind nun ohne Kühlkörper in dem Temperaturbereich den wir vorher mit Kühlkörper ereicht haben.

Die veränderte Wärmeentwicklung ist gut auf den Thermobildern zu sehen.

Vorher:



Nachher:



ROM auf EPROM

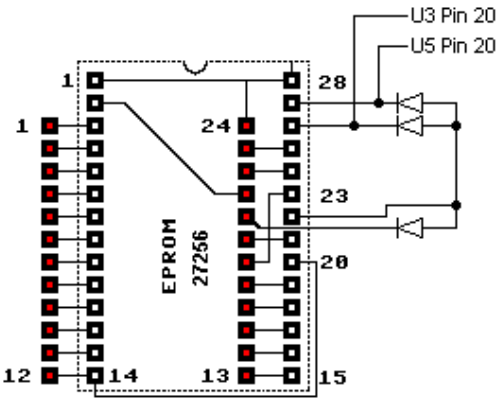
Die ROM Bausteine für das Kernel, Basic, und Zeichensatz benötigen je ca. 65 mA.
Ein EPROM 27CXXX hat lediglich eine Stromaufnahme von 4-7 mA.
Durch das ersetzen der drei ROM bausteine durch ein EPROM kann der Stromverbrauch auf 5 Volt Netzteilschiene um bis zu
270 mA reduziert werden, bei Verwendung eines 27C512 Eprom ist auch ein Kernel/Basic/Chear Umschaltung möglich.
Zuerst müssen die ROM Bausteine ausgelötet werden, und IC Präzisions IC Sockel verbaut werden.
Anschließend wird in den Mittleren Sockel U4 der Adaptersockel gesteckt und die beiden Chip Select leitungen auf Pin 20 des Nachbarsockels gesteckt

Speicheraufteilung im EPROM

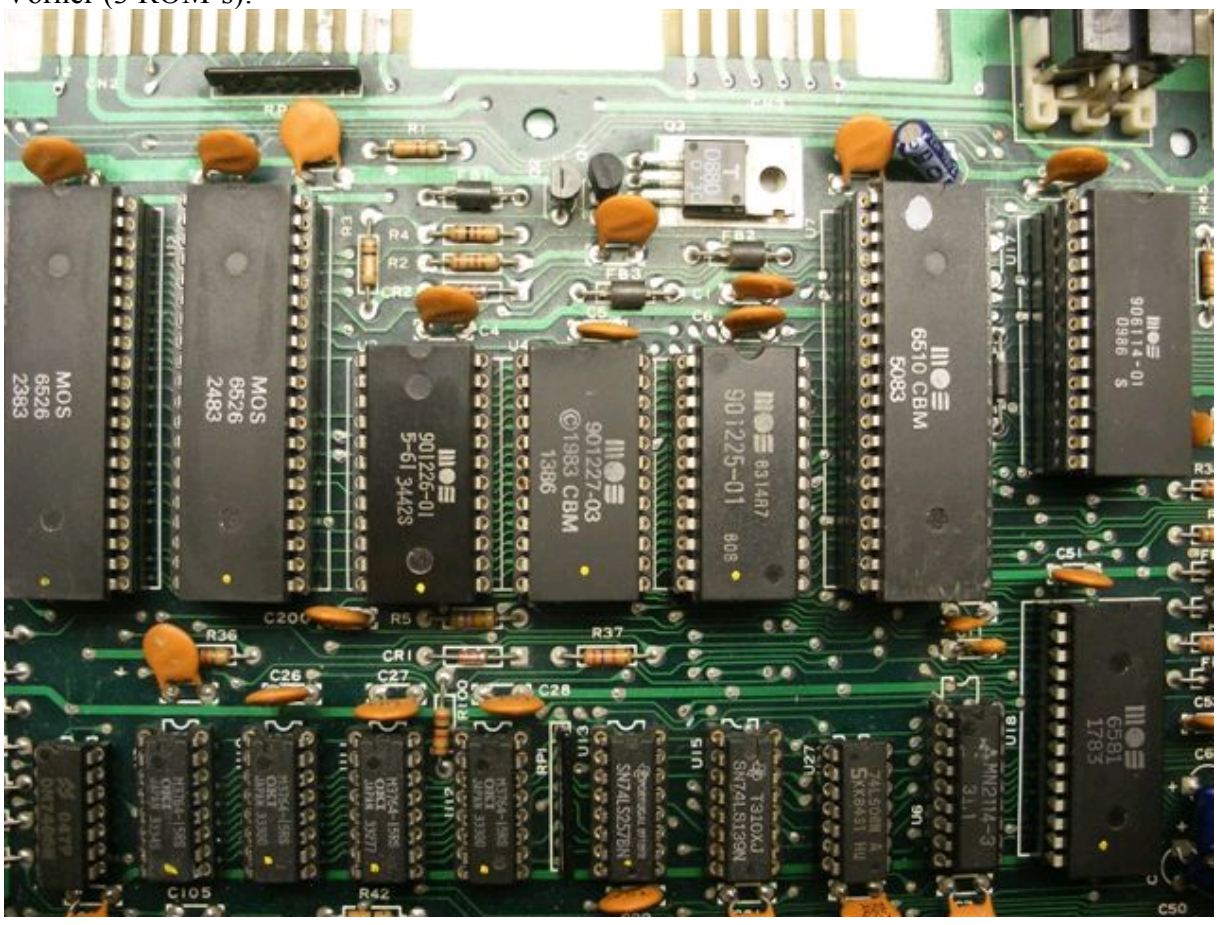
```
--------------------------------32k
Kernel   6000 bis    7FFF
--------------------------------24k
Free     5000 bis    5FFF
--------------------------------20k
Char     4000 bis    4FFF
--------------------------------16k
Basic    2000 bis    3FFF
--------------------------------8k
Free     0000 bis    1FFF
--------------------------------0
```

Benötige Bauteile:
4 * IC Sockel 24 Pol.
     IC Sockel 28 Pol.
3*   Diode SB-140
EPROM 27C256

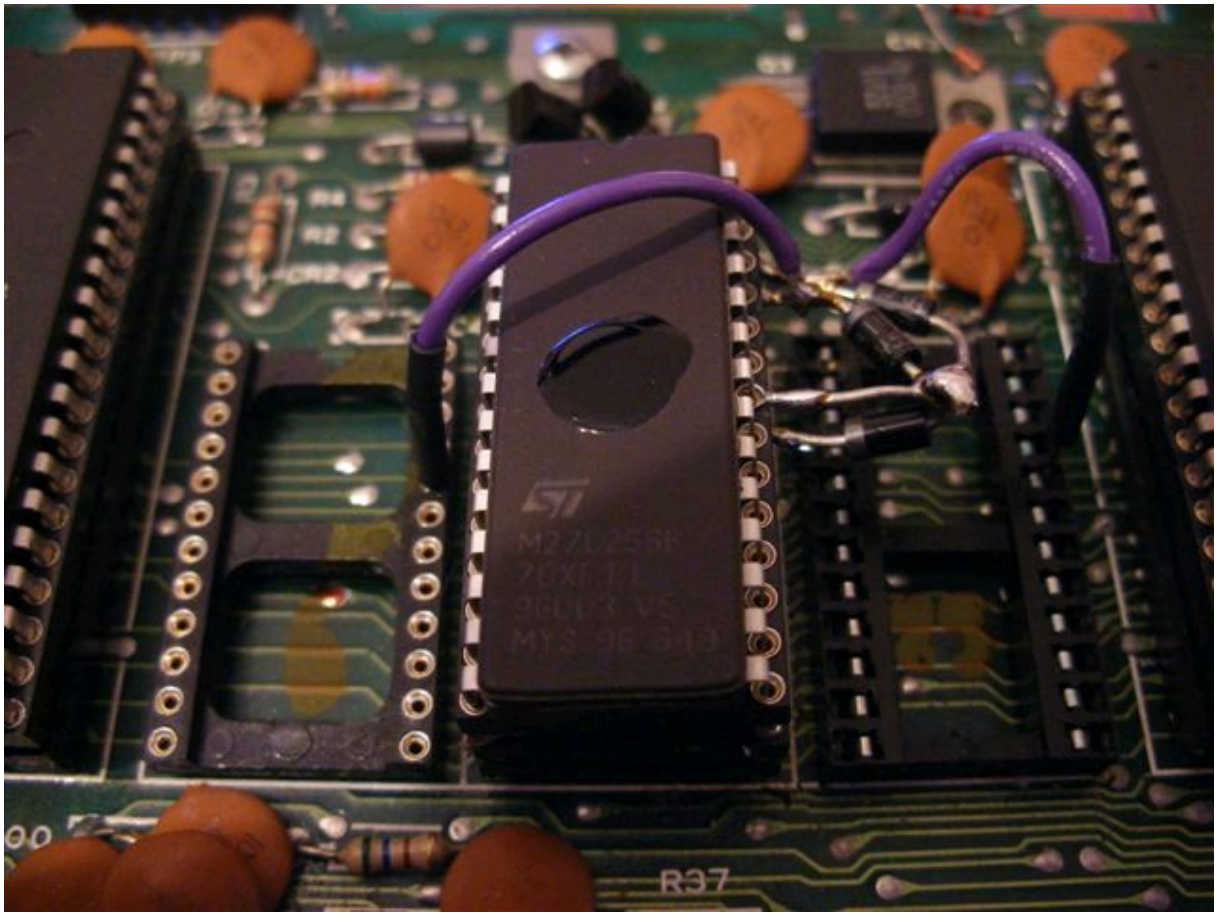
Schaltbild:



Vorher (3 ROM’s):



Nachher:



Ersatz-PLA

Die PLA im C64 zieht ca. 100mA, ein Umbau auf ein 27C512 Eprom reduziert die Stromaufnahme auf ca. 6mA.

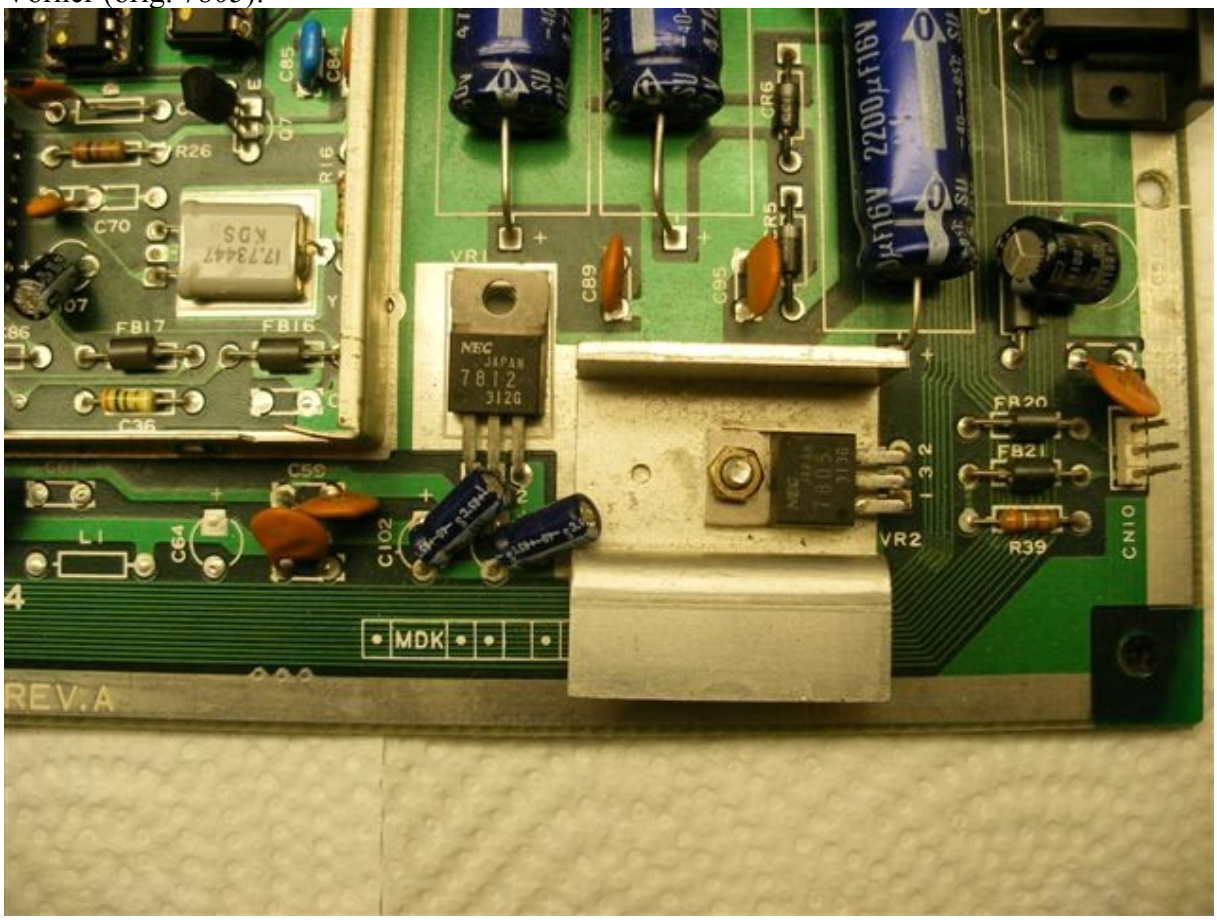
Es wird zuerst die PLA ausgelötet, und ein IC Sockel eingelötet.
Aus zwei IC Sockeln wird der EPROM-PLA Adapter zusammengelötet, und das gebrannte EPROM eingesetzt, fertig. Eine detaillierte Anleitung gibt hier im entsprechenden Kapitel.

7805 auf Schaltregler

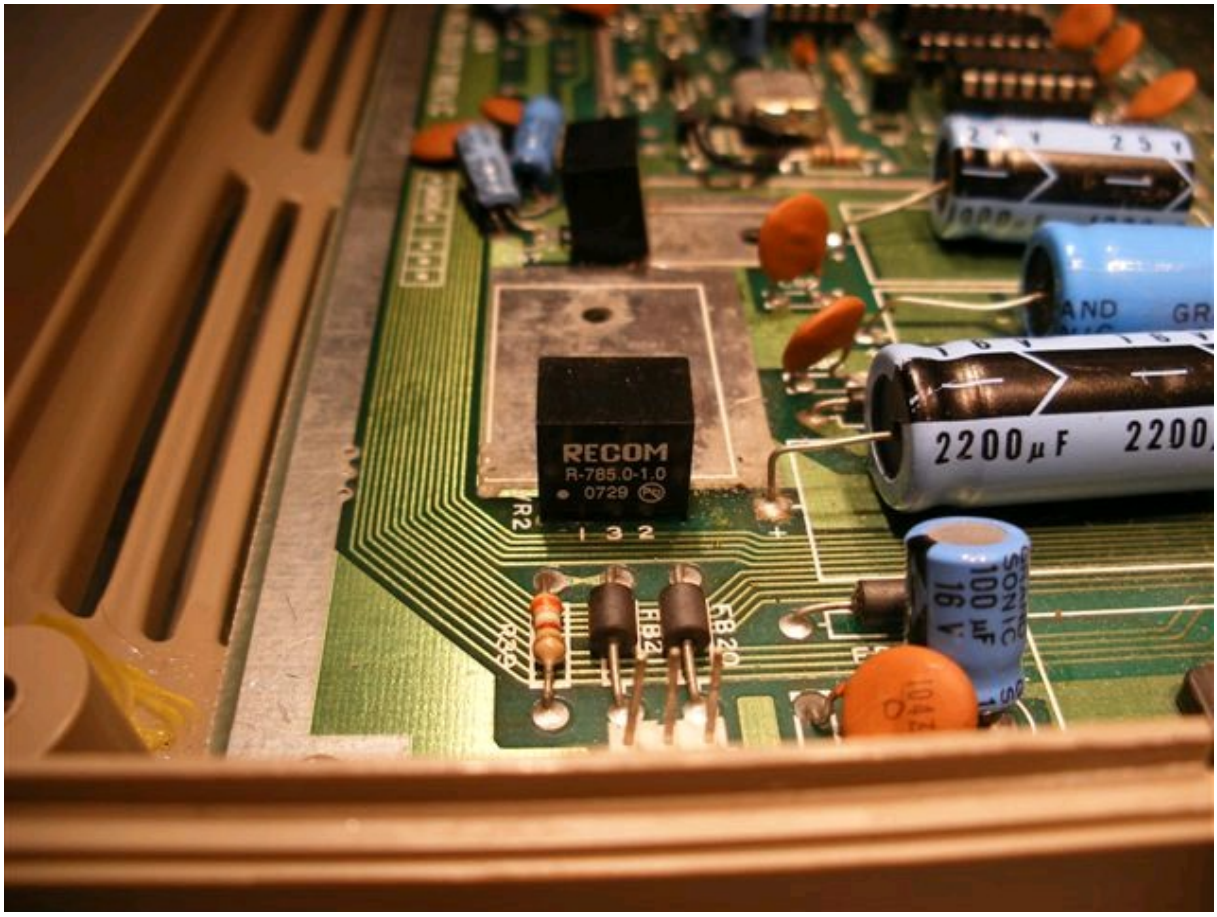
Der 7805 Spannungsregler auf der ASSY 250407 wird mit einem Strom von ca.300 mA belastet, die Eingangsspannung des Reglers sind die gleichgerichteten 9 Volt vom Netzteil. Der Regler hat nach der Glättung der 9 Volt ca. 10 Volt Eingangsspannung bei einer Ausgangspannung von 5 Volt.
Der Wirkungsgrad des Reglers ist 10 Volt - 5 Volt = 5 Volt Spannungsabfall am Regler, 5 V : 10V =50% Wirkungsgrad.
Wenn der 7805 gegen einen Pinkompatiblen Schaltregler ausgetauscht wird, steigt der Wirkungsgrad auf bis zu 94 %.
Conrad Elektronik hat zum Beispiel die 154496 - 62 im Angebot, einfach die Schraube am Kühlkörper Lösen, den Regler auslöten und den neuen Schaltregler einsetzen.

Benötigte Bauteile:
Schaltregler Kosten ca. 10,16 Euro

Vorher (orig. 7805):



Nachher:

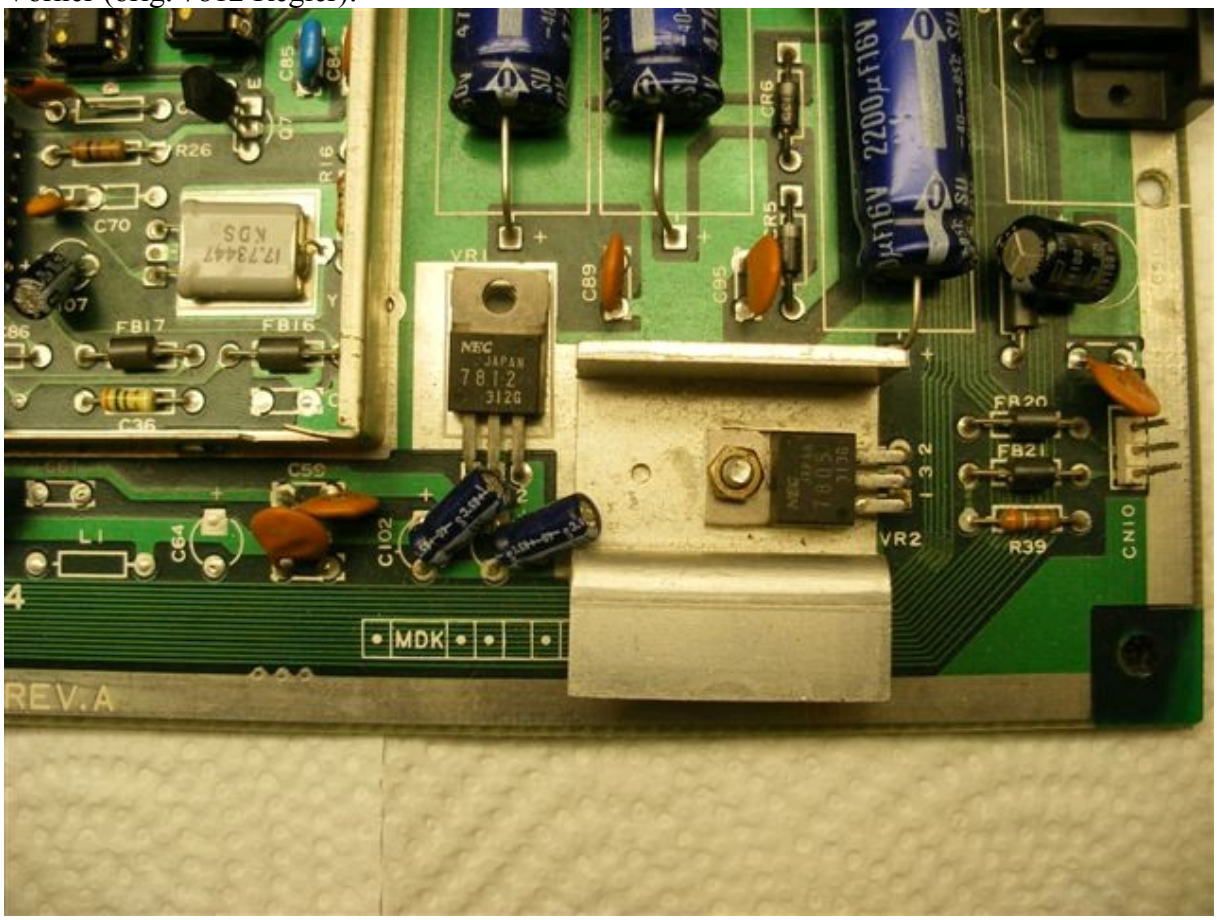


7812 auf Schaltregler

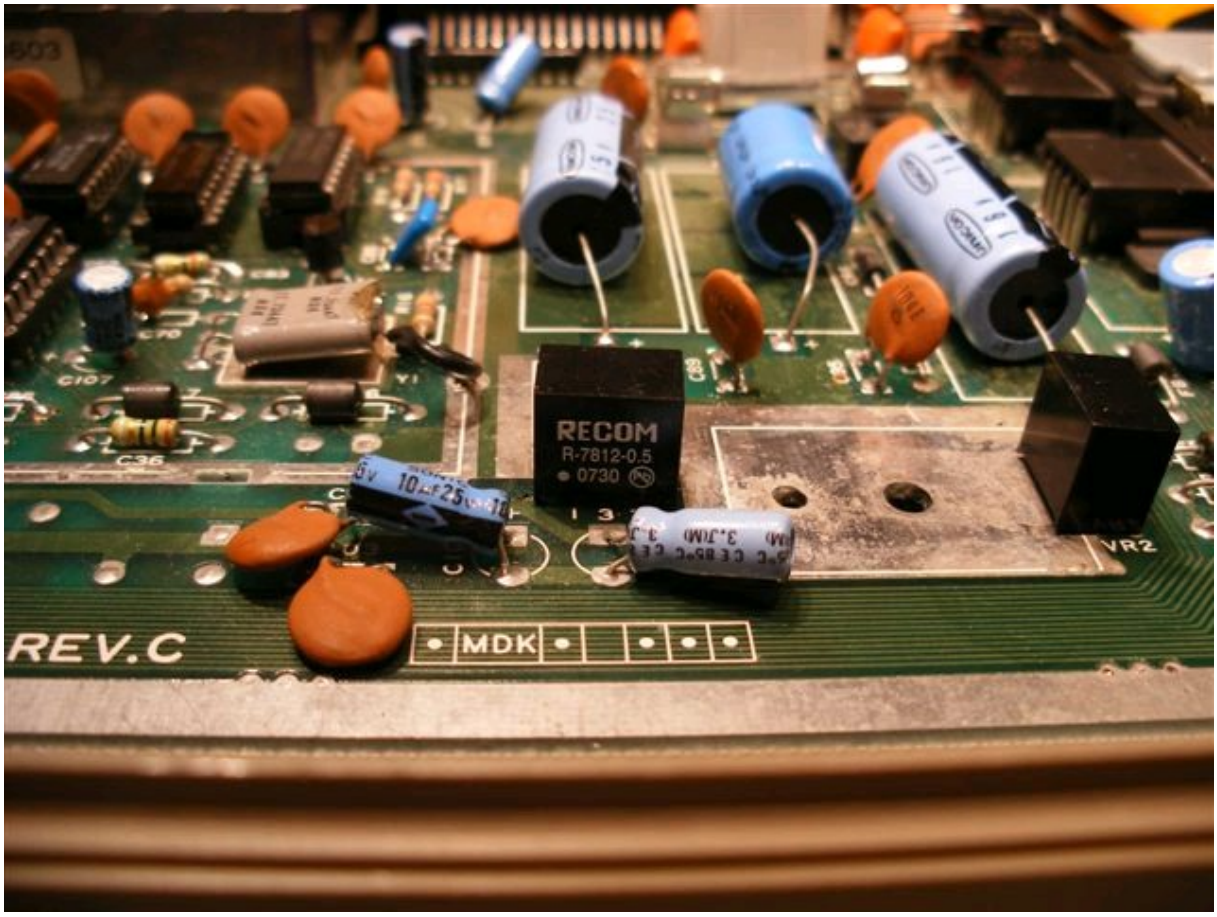
Bei dem 7812 auf der ASSY 250407 gilt das gleiche wie bei dem 7805 Regler..
Strom ca. 100 mA bei 12 Volt, Eingangsspannung 17 Volt (Spannungsverdopplerschaltung mit Kondensator und Gleichrichter) Wirkungsgrad ca. 70%.
Durch Wechsel des Gleichrichters (Siehe 6.) und Reduzierung der Stromlast auf der 9 Volt Schiene erhöht sich die Spannung am Eingang auf ca. 21 Volt, was den Wirkungsgrad des Schaltreglers zu Gute kommt (Bis zu 97%)

Benötigte Bauteile:
Schaltregler Kosten ca. 10,16 Euro Bestellnummer 154484 – 62

Vorher (orig. 7812 Regler):



Nachher:



RAM Umbau von 4164 auf 53c464-70

Der Arbeitsspeicher des C64 mit einer ASSY 250407 hat acht Bausteine a 64k*1, jeder Baustein benötigt zwischen 16-21 mA. Das bedeutet eine Gesamtstromaufnahme von 130 mA bis 170 mA.

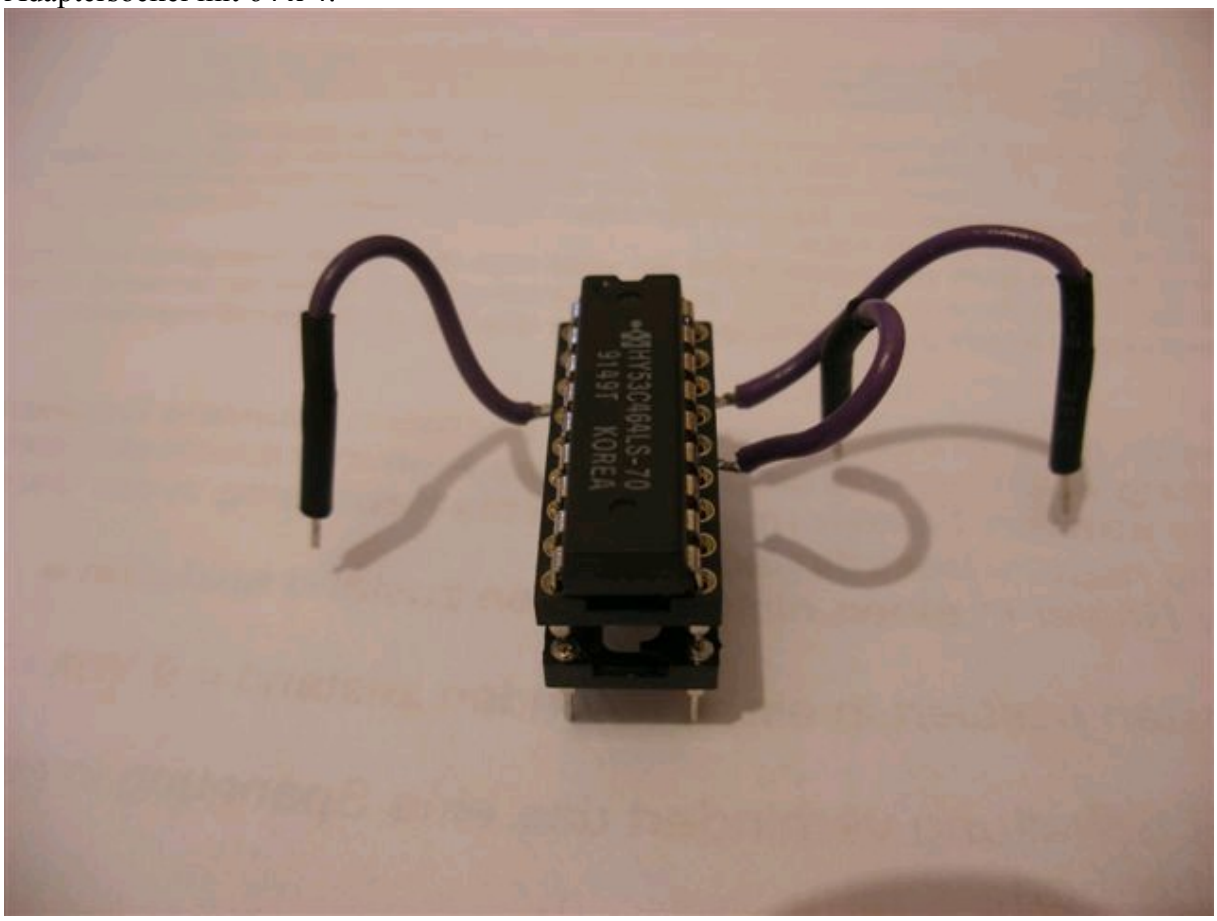
COMMODORE hat ja schon bei der ASSY 250466 gezeigt das ein Umbau von 4164 auf 41464 möglich ist. Zuerst müssen alle acht RAM Bausteine ausgelötet werden, die leeren Plätze mit 16 Pol. IC Sockeln bestückt werden.

Nach dem Schaltplan aus zwei IC Sockeln die RAM Adapter bauen.
Die 53C464-70 haben die geringste mir bekannte Stromaufnahme, ein -100 oder normaler 41464 braucht ca. 5-10mA mehr. Zu beziehen bei http://www.sh-halbleiter.de

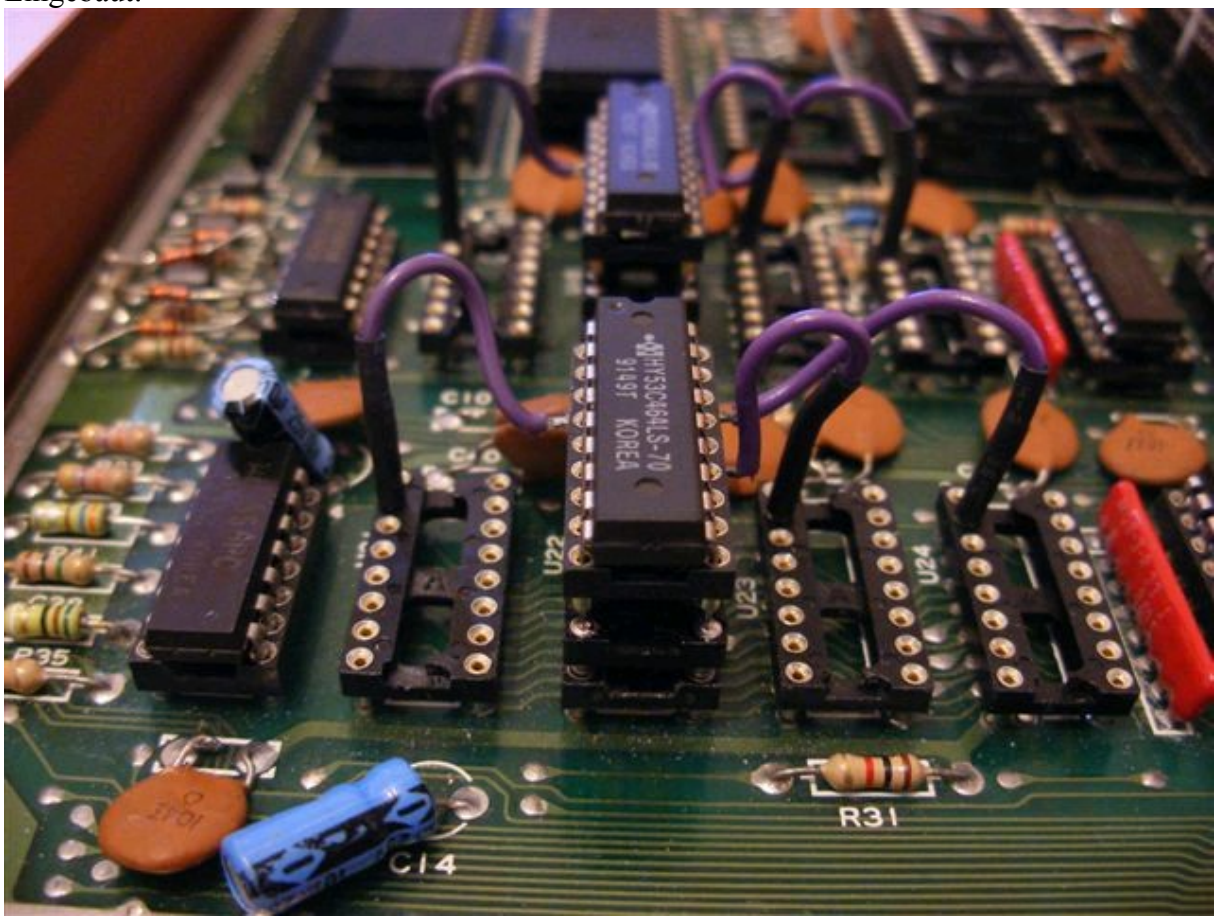
Der Adapter sollte auf die Postion U10 und U22 gesteckt werden, die Anschlussleitungen zu den leeren IC Sockeln sollten zu kurz wie möglich sein.

Stromersparnis des Umbaus ca. 90 bis 140 mA

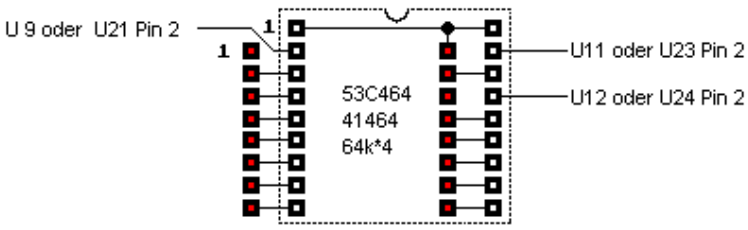
Adaptersockel mit 64 x 4:



Eingebaut:



Schaltplan:



Gleichrichter auf Schottky Dioden

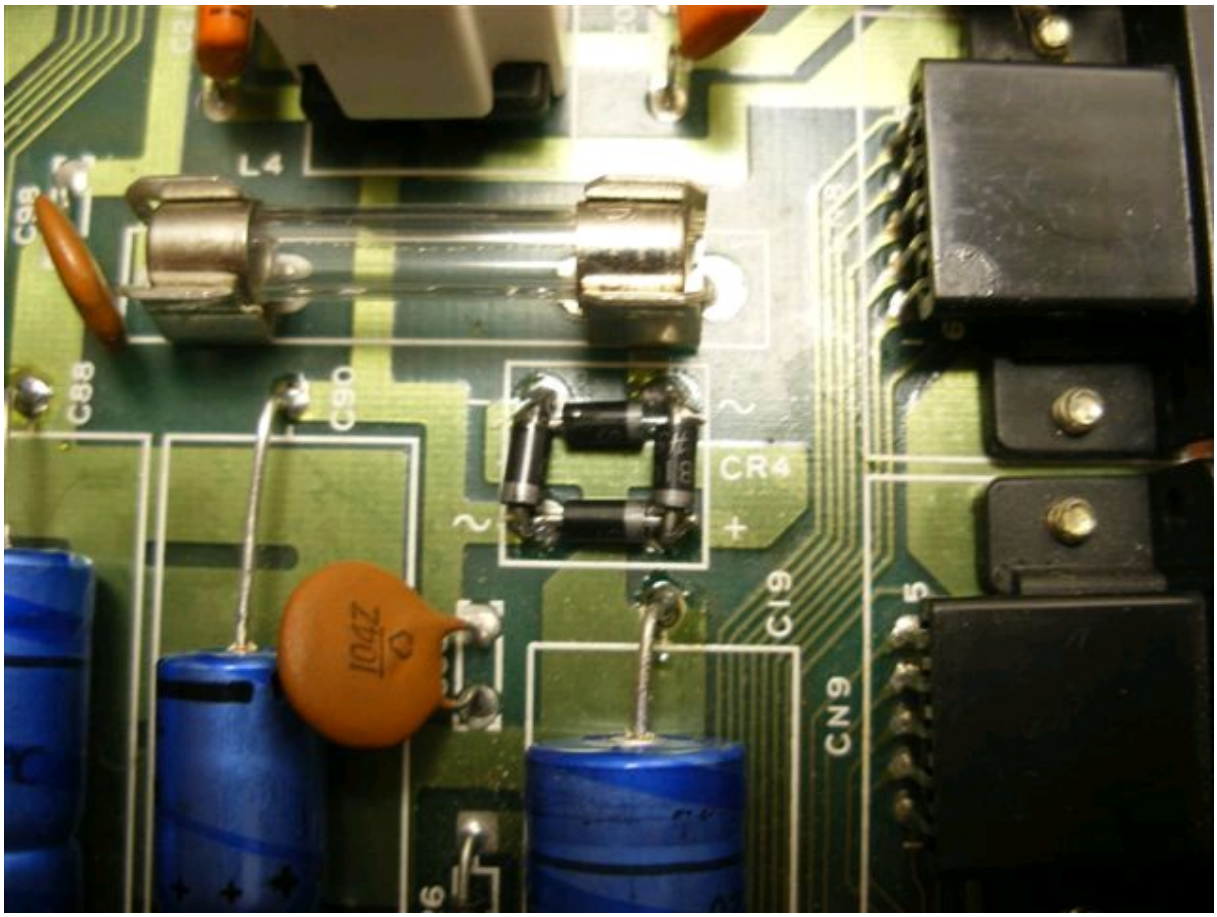
Der Gleichrichter für die 9 Volt im Bereich des Netzschalters ist auch eine Wärmequelle. Spannungsabfall ca.2* 0,7 Volt bei ca. 600mA Last = 0,84 Watt

Bei Schottky Dioden reduziert sich der Spannungsabfall auf ca.2*0,3 Volt = 0,36 Watt

Empfohlener Schottky Dioden Typ: SB140

Der Umbau des Gleichrichters ist nur sinnvoll, wenn auch die Regler 7805 und 7912 umgebaut wurden! Ansonsten werden die 0,38 Watt bei den Reglern nutzlos in Wärme umgesetzt.

Am Besten den Alten Gleichrichter mit Entötsaugpumpe auslöten, die neuen Dioden passend zurechtbiegen und in die Gleichrichter Bohrungen stecken und verlöten, siehe Bild.



Kühlkörper nachrüsten

Um die Wärme an den kritischen IC's abzuleiten haben sich Kühlkörper bewährt.
Die Kontaktfläche zwischen Kühlkörper und IC-Gehäuse ist der kritischste Punkt.
Der Wärmewiderstand sollte möglichst gering sein, Unebenheiten zwischen Kühlkörper und IC müssen ausgeglichen werden, und der Kühlkörper soll am IC haften.

Normale Wärmeleitpasten erfüllen den Zweck der Wärmeleitung und des Ausgleichs sehr gut, sind aber nicht für die Fixierung des Kühlkörpers geeignet.

Da das IC Gehäuse aus zwei Komponenten Epoxyd Harz hergestellt wurde, bietet sich natürlich der gleiche Werkstoff auch für die Befestigung des Kühlkörpers an. Die schlechtere Wärmeleitfähigkeit ist bei der geringen Wärmeleistung und Schichtdicke zu vernachlässigen.

Als Vorbereitung ein Foto von der Leiterplatte machen, so ist ggf. nachträglich noch feststellbar wo welches IC verbaut ist.

Mit einem wasserfesten Stift bei der IC-Kerbe eine Markierung auf die Seite machen, da bei verklebtem Kühlkörper die Kerbe nicht mehr ersichtlich ist, wenn das IC gesockelt ist lohnt es sich auch die Typbezeichnung auf die Unterseite zu schreiben

Für die eigentliche Verklebung werden nur 2-4 Tropfen 2K Kleber benötigt. Man sollte aber mehr anmischen weil so das Mischungsverhältnis genauer eingehalten werden kann.
Die Klebstofftropfen mittig auf die Längsachse aufgetragen und der Kühlkörper auflegen, und vorsichtig mit einem Hilfsgewicht angepressen. Überflüssiger Klebstoff darf nicht an den IC-Beinen herunter laufen, überflüssigen Kleber im Notfall mit einem Q-Tip entfernen. Der Klebstoff härtet schneller aus wenn man ihn auf ca.40 °C erwärmt.

Vor der Montage solle man auch prüfen ob der Kühlkörper genug Platz hat!

C64 – Netzteilschutzschaltung

Von Peter Trachte (AREA51HT).

Bei den inzwischen 25 Jahren alten Netzeilen kommt es leider immer häufiger zu Defekten. Problematisch ist dabei die Unterdimensionierung von Bauteilen. Hier insbesondere die Gleichrichterdioden (besser 3A Dioden als 1A) und die Spannungs- und Temperaturbeständigkeit des Sieb-Elko's (anstatt 4700uF/16V besser 4700uF/25-35V und 105/125 °Grad Celsius Temperaturbeständigkeit).

Im forum64 ist ein solcher Umbau beschrieben. An dieser Stelle möchte ich diesen nicht wiedergeben, da der Umbau doch etwas komplizierter ist und direkt mit Netzspannungs-berührenden Teilen gearbeitet wird.

Stattdessen sollte man in Betracht ziehen, sich ein komplett neues Netzteil bauen zu lassen.

Um aber den C64 vor einem Netzteilausfall und der dann unter Umständen fließenden Überspannung auf der 5V Zuleitung zu schützen soll hier eine entsprechende Schutzschaltung vorgestellt werden.

Eigenschaften der Schutzschaltung:

1. Negative Spannungen werden kurzgeschlossen
2. Spannungsdrifts durch langsame Ausfälle sollen erkannt und durch Notabschaltung des Netzteiles gestoppt werden
3. Kurze Spikes (Spannungstransienten) sollen abgefangen werden

Als Grundlage der Schaltung wurde eine Crowbar (Brechstange) Schaltung verwendet.

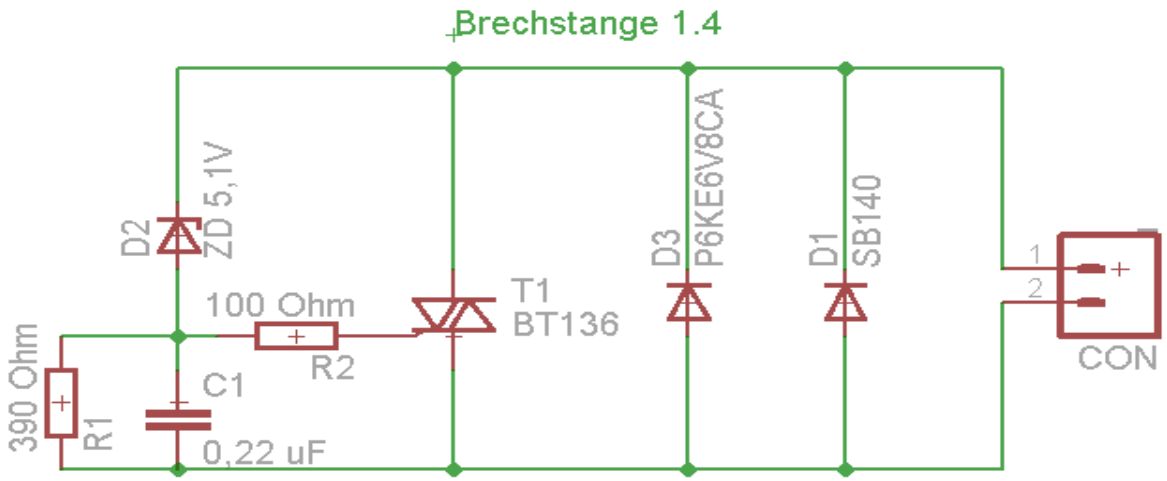
Einfaches Konzept:
Wenn die Spannung über einen gewissen Wert steigt, schließt die Schaltung die Spannung einfach kurz. Dabei soll die zusätzliche Gerätesicherung im Sekundärkreis auslösen (Schmelzsicherung) und die teure Schaltung und der C64 sind gerettet.

Um vor negativen Spannungen im Verpolungsfall zu schützen hat es sich bewährt, eine Leistungsdiode mit der Kathode (Ring) nach Plus in den Spannungseingang zu schalten. Im Fehlerfall wird die Spannung so ebenfalls kurzgeschlossen.

Die Erfahrung zeigt, das die meisten Halbleiter ab ca. -0,7 bis -0,9 zerstört werden, deswegen kommt eine SB140 zum Einsatz, die unter Belastung ca. 0,3 bis 0,5 Volt Spannungsabfall hat, und sehr schnell reagiert.

Um vor Spannungsspitzen, die durch ESD Endladung, Bitzeinschläge in der Nähe, oder auch Netzstörungen in die 5 Volt gelangen zu schützen, kommt eine Suppressordiode zum Einsatz. Diese Diode ist eine Zenerdiode mit besonders steiler Kennlinie und sehr hoher Impulsleistung. Die verwendete Diode kann bis zu 600W Peak vertragen und durch die bidirektionale Ausführung ist auch der Verbau verpolungssicher.

Schaltplan:



Nachteil der Schaltung ist, das sie abgeglichen werden muss. Die verwendete Zenerdiode hat eine große Toleranz, so das eine 4,7 Volt oder auch eine 5,1 Volt in Frage kommt, am besten beide Typen beschaffen. An einem Labornetzteil 5,00 Volt und die Strombegrenzung auf 500mA einstellen. Die Schutzschaltung anschließen und den Strom beachten, es sollten weniger als 5mA fließen. Nun langsam die Spannung erhöhen bis die Crowbar auslöst und die Spannung unter 1 Volt zusammenbricht. Bei 20°C sollte das bei ca. 5,3 bis 5,5 Volt geschehen. Wenn die Schaltung zu früh auslöst kann der Wiederstand R1 verkleinert werden (Unteres Limit ca. 180 Ohm), oder eine andere Z-Diode (z.B 5,1V) getestet werden.
Bei zu später Auslösung kann R1 auf ca. max 560 Ohm vergrößert werden oder eine andere Z-Diode getestet werden. Vergrößern von C1 reduziert die Empfindlichkeit des Ansprechens der Schaltung.

Da die Bauteile, insbesondere die Z-Diode Temperatur-Drift haben ist auch ein Test unter erhöhten Temperaturen notwendig. Mit einem Haarfön oder besser einem regelbaren Heißluftfön die Schaltung auf ca.50-60°C erwärmen und den Versuch wiederholen.
Die Auslösespannung darf nicht unter 5,25 absinken.

Wenn die Schaltung diese Tests überstanden hat, kann sie z.B einfach in einen C64 verbaut werden. Dabei wird +5V und GND/Masse hinter dem Ein-/Ausschalter mit der Schutzschaltung **über eine zusätzliche, superflinke Schmelz-Sicherung (0,8-1,0A)** verbunden. Erst über diese Zusatzsicherung im Sekundärkreis, darf der C64 dann mit +5V versorgt werden! Im Störungsfall löst dann diese Sicherung aus.

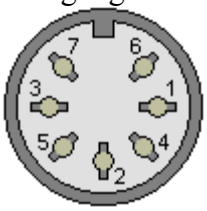
Der Betrieb mit einem orig. C64 Netzteil empfiehlt sich nicht! Das Netzteil geht unter Umständen in einen Zustand, wo keine Sicherung auslöst, max. Last an Strom abgegeben wird und sich das Netzteil dabei bis zum schmelzen (Brandgefahr!) aufheizt. Generell alles immer nur hinter einer schaltbaren Steckdosenleiste unter Aufsicht betreiben und im Störfall sofort manuell die Stromzufuhr abschalten.

Netzteil nur für C64G/-II / ASSY 250469 und VC20

Dem VC20 mit dem C64 Netzteilanschluß reichen +5V zum Betrieb. Die 9V~ werden ggf. nur für Zusatzmodule benötigt. Daher kann man hier ein modernes 5V Schaltnetzteil verwenden. Diese gibt es z.B. als Restposten günstig zu erwerben.

Ich habe folgendes Netzteil von Pollin verwendet: Bestell-Nr.: 350765
Es ist schön klein und liefert 2A an 5V. Kostet nur 4,95€.

Belegung der Netzsteilbuchse am C64:



Pin	Signal	Bemerkung
1	GND	Masse
2	GND	Masse
3	GND	Masse
4	-	unbelegt, evtl. auch 5 Volt
5	+ 5V DC	5 Volt Gleichspannung
6	9V AC	9 Volt Wechselspannung
7	9V AC	9 Volt Wechselspannung

Ein C64G (ASSY: 250469) arbeitet so auch schon – allerdings ohne Sound, da der 8580 SID über die 9V~, Gleichrichter und 7809 Spannungsregler intern mit +9V versorgt wird.

Abhilfe kann hier aber eine SID Ersatzschaltung (siehe SwinSID) schaffen! Damit hat man am C64G auch Sound, obwohl man den Rechner nur mit +5V versorgt.

Bei den anderen C64 Platinenrevisionen reichen nur +5V als Stromversorgung leider nicht. Hier muß man noch die 9V~/ca. 1A über einen separaten Trafo zur Verfügung stellen.

Bitte immer bei Arbeiten mit Netzspannung die Sicherheitsbestimmungen beachten! Natürlich gehört obiges Netzteil in ein Kunstoffgehäuse und in den Primärkreis eine Schmelzsicherung.

Ich empfehle grundsätzlich den Betrieb von selbstgebauten Schaltungen und Retro-Hardware nur hinter einer schaltbaren Steckdosenleiste und nur unter Aufsicht.

PC Netzteil Adapter

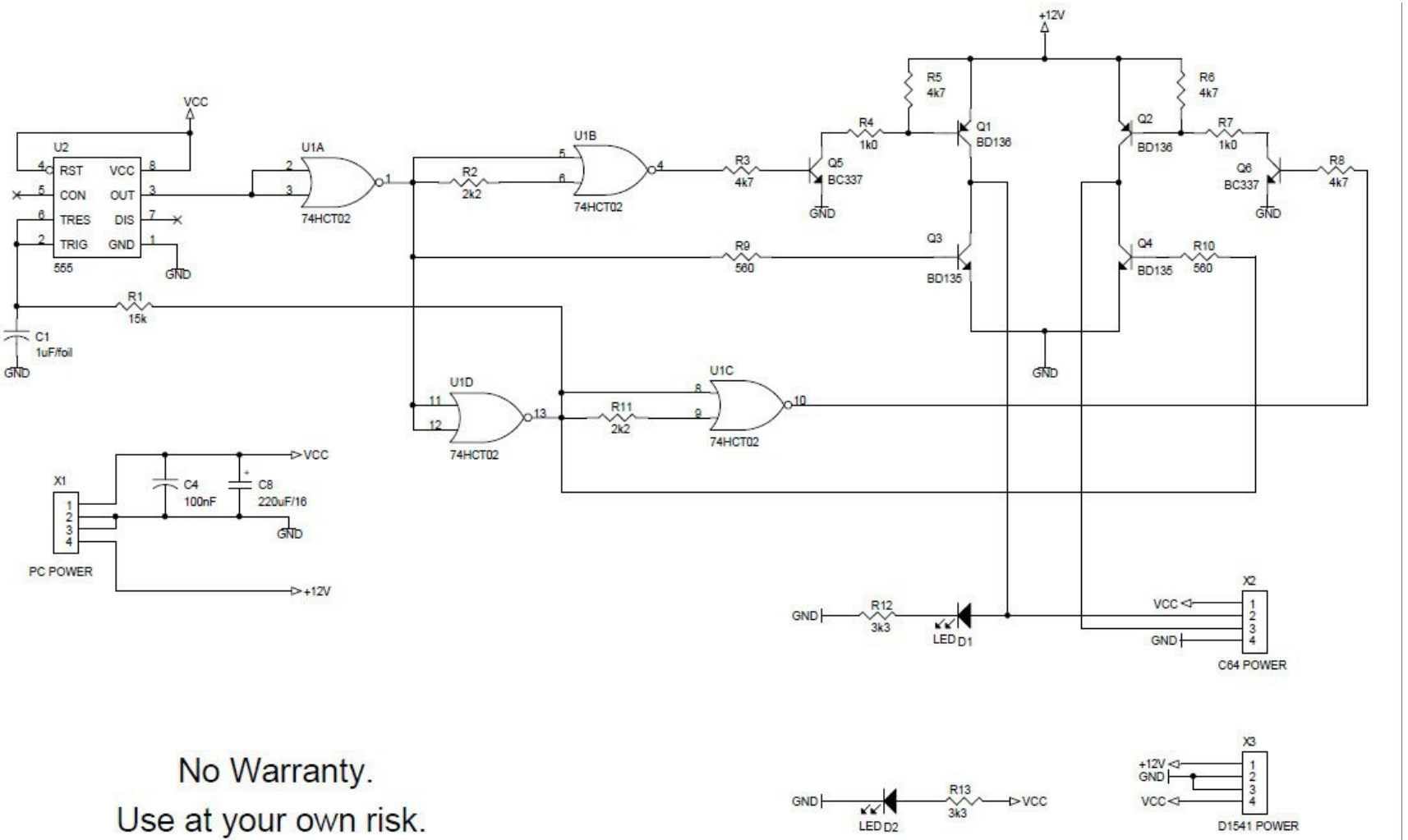
Die AC 9V POWER lösung ist von: Tomas Pribyl <tomas.pribyl@ide64.org>
Bilder von: Josef Soucek <josef.soucek@ide64.org>
Mit freundlicher Genehmingung von Josef Soucek [josef.soucek@ide64.org]

Link: http://ide64.org/stuff.html

Die Schaltung gwinnt aus +5V und +12V DC, die für den C64 benötigten +5V DC und ~9V AC.

Achtung: Der C64 sollten dann am PC Netzteil oder einem Steckdosenschalter ein- und ausgeschaltet werden! Am C64 reicht nicht mehr, da die 9V AC und die +5V eine gemeinsame Masse haben. Mit dem original C64 Einschalter läge so noch eine 9V Phase am Gerät an.

Auch hier sind ausreichende Erfahrungen mit elektrischen Schaltungen unbedingt erforderlich. Aufbau und Betrieb auf eigene Varantwortung.



No Warranty.

Use at your own risk.

http://www.volny.cz/dundera

Tomáš Přibyl dundera@volny.cz		
Title: C64 - Power Supply		
Size: A4	Document Number: Power1	Rev: 1

Basic und Kernal durch Eprom ersetzen bei ASSY 250469

Beim C64 mit ASSY 250469 sind Basic und Kernal ROM in einem 23128 DIL-28 ROM Baustein kombiniert. Die Bezeichnung ist: **251913-01**

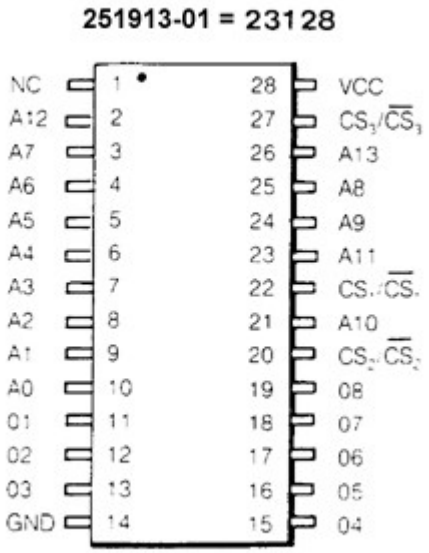
Dieser Baustein kann ohne jeden Adapter durch ein 27(C)128 Eprom ersetzt werden, da die beiden Bausteine pinkompatibel sind.

Ein Eprom Image kann unter Windows einfach aus z.B den bei dem Vice Emulator beiliegenden Basic und Kernal Roms erzeugt werden. Dazu in einer Kommandozeile eingeben:

copy /b basic + kernal c64c.rom

Wichtig ist der Schalter '/b' für binärer Kopieren (keinerlei Ändern von Bytes). ‚basic' und ‚kernal' sind die beiden 8192 Bytes großen Basic und Kernal ROM's. ‚c64c.rom' ist das 16384 Bytes große, resultierendes Eprom Image, welches dann in einem 27(C)128 gebrannt werden muß.

Den orig. ROM einfach auslöten, eine 28-pol DIL Fassung einlöten und gebranntes Eprom einstecken und fertig.

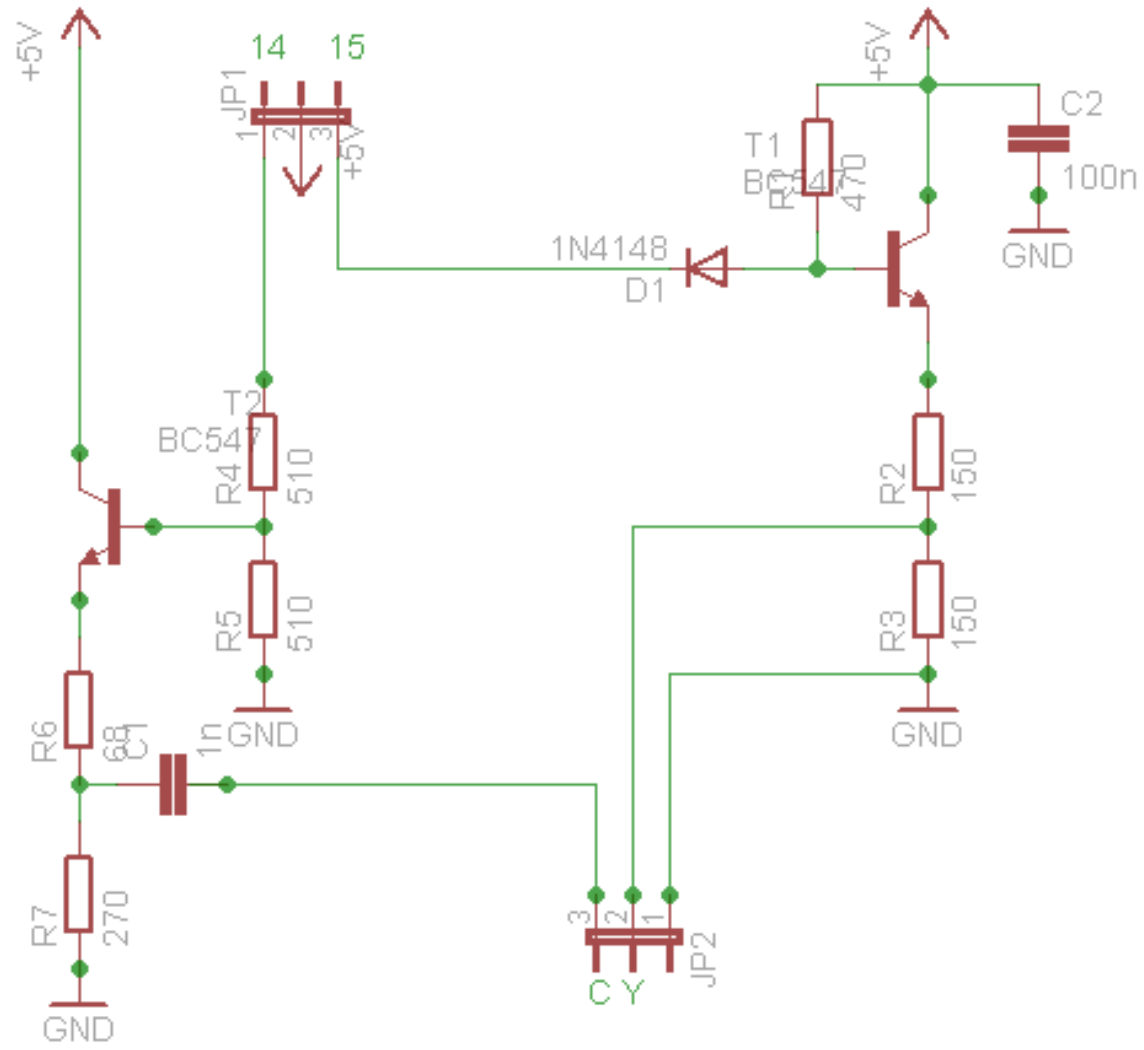


S-Videoausgang am C64

Schaltung von Björn 'JMP$FCE2' Wieck

Was als Diskussion im Forum64 begann, den TV-Modulator auf der Assy 250469 zu entfernen, um Platz für z.B sd2iec etc. zu haben, führte letzendlich zu einer Schaltung, die getrennte Chroma und Luma+Sync (S-Video) Signale zur Verfügung stellt. Diese Signale führen insbesondere bei TFT-Bildschirmen/Fernsehern zu einer besseren Bildqualität als das FBAS Signals des C64. An Bauteilen werden nur zwei Transistoren Typ: BC547 (A-C), zwei Kondensatoren, eine Diode und ein paar Widerstände benötigt. Zwei Eingangssignale müssen direkt am VIC Chip Pin 14 und Pin 15 abgegriffen werden. Dazu z.B. die beiden Pin's des VIC's aus der Fassung lassen und so hochbiegen, das keine Verbindung zur Hauptplatine mehr besteht und man Kabel anlöten kann. Der FBAS Ausgang funktioniert dann nicht mehr. Die Schaltung läßt sich einfach auf Lochraster oder einer einseitigen Platine aufbauen.

Schaltplan:



Floppy 1541 - Wartungs- und Einstellarbeiten

Von Björn 'JMP$FCE2' Wieck

Hier wird gezeigt, wie eine 1541 Generalüberholt und auf aktuelle Netzspannung umgerüstet wird. Es geht hier nicht darum eine defekte 1541 zu reparieren. Es wird für keinerlei Schäden die auftreten könnten gehaftet, sei es durch unsachgemäßen Gebrauch der Messgeräte oder Werkzeuge oder Schäden die durch ESD auftreten können.

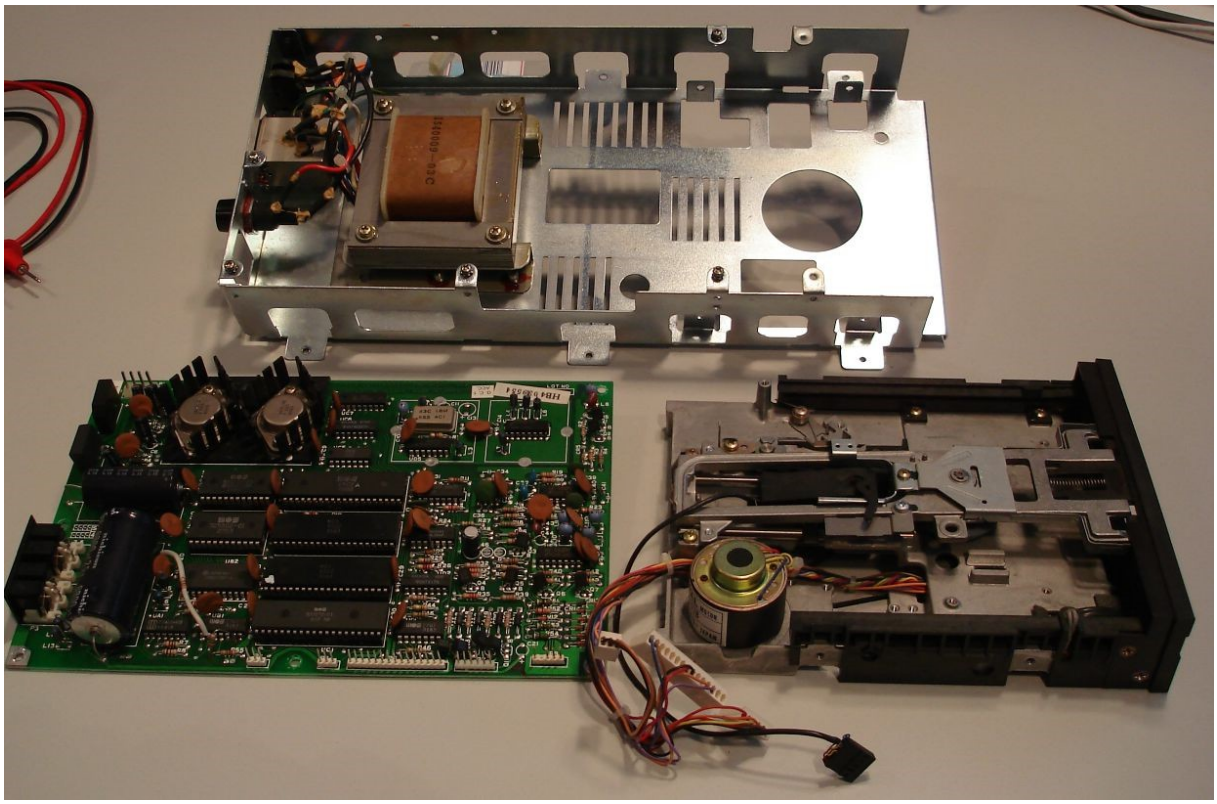
Es wird benötigt:

- Schraubendreher Kreuzschlitz PH1
- Schraubendreher Kreuzschlitz PH2
- Schraubendreher Schlitz 2mm
- Multimeter
- Lötkolben
- Schrumpfschlauch
- Oszilloskop (dieses nur für die Spurlageneinstellung)
- Q-Tips
- Küchentücher
- 2-Euro Münze + etwas Garn
- etwas starrer 0,6mm Draht
- Pinsel
- Staubsauger
- säurefreies nicht harzendes Öl (z. B. Ballistol)
- Isopropylalkohol 99%
- Original 1541 Test/Demo Diskette
- Software: Kwik Copy
- Software: Floppy Adjust

Ein Tip vorweg:

ESD ist ein ernstzunehmendes Thema und für 60% aller Schäden an alter Commodore-Hardware verantwortlich. Wer keinen ESD-Konformen Arbeitsplatz hat sollte bitte nach Möglichkeit während der Arbeiten keine Kleidung mit hohem Kunststoffanteil und entweder (falls vorhanden) Schuhe mit leitfähiger Sohle oder sonst keine Schuhe tragen.

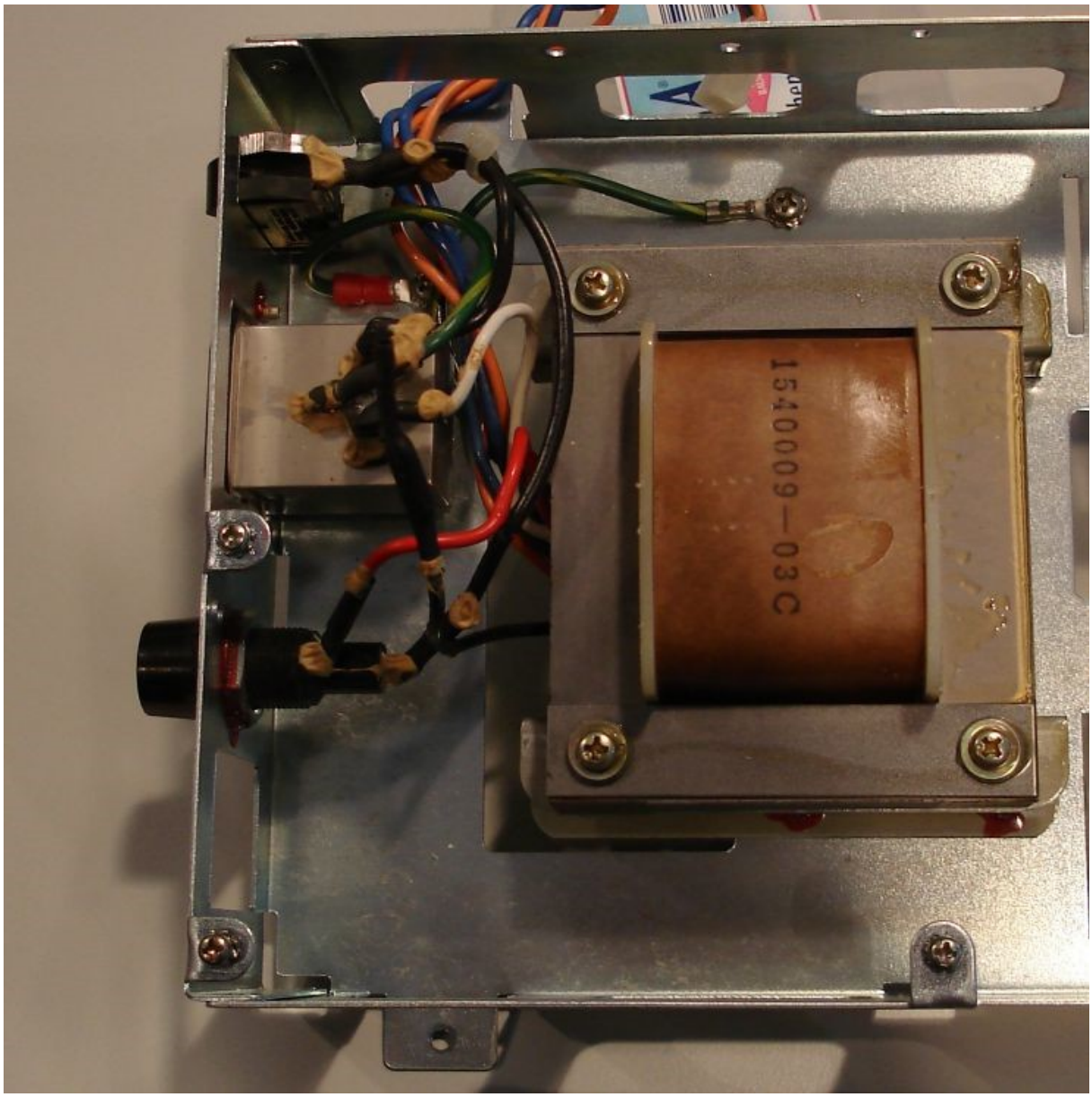
Zuerst wird das Gehäuse geöffnet und mit dem Staubsauger und Pinsel sämtlicher Staub entfernt. Dann werden alle Steckverbinder gelöst, die Platine demontiert und das Chassis aus dem Gehäuseunterteil entfernt. Zum Schluss wird das Laufwerk aus dem Chassis geschraubt.



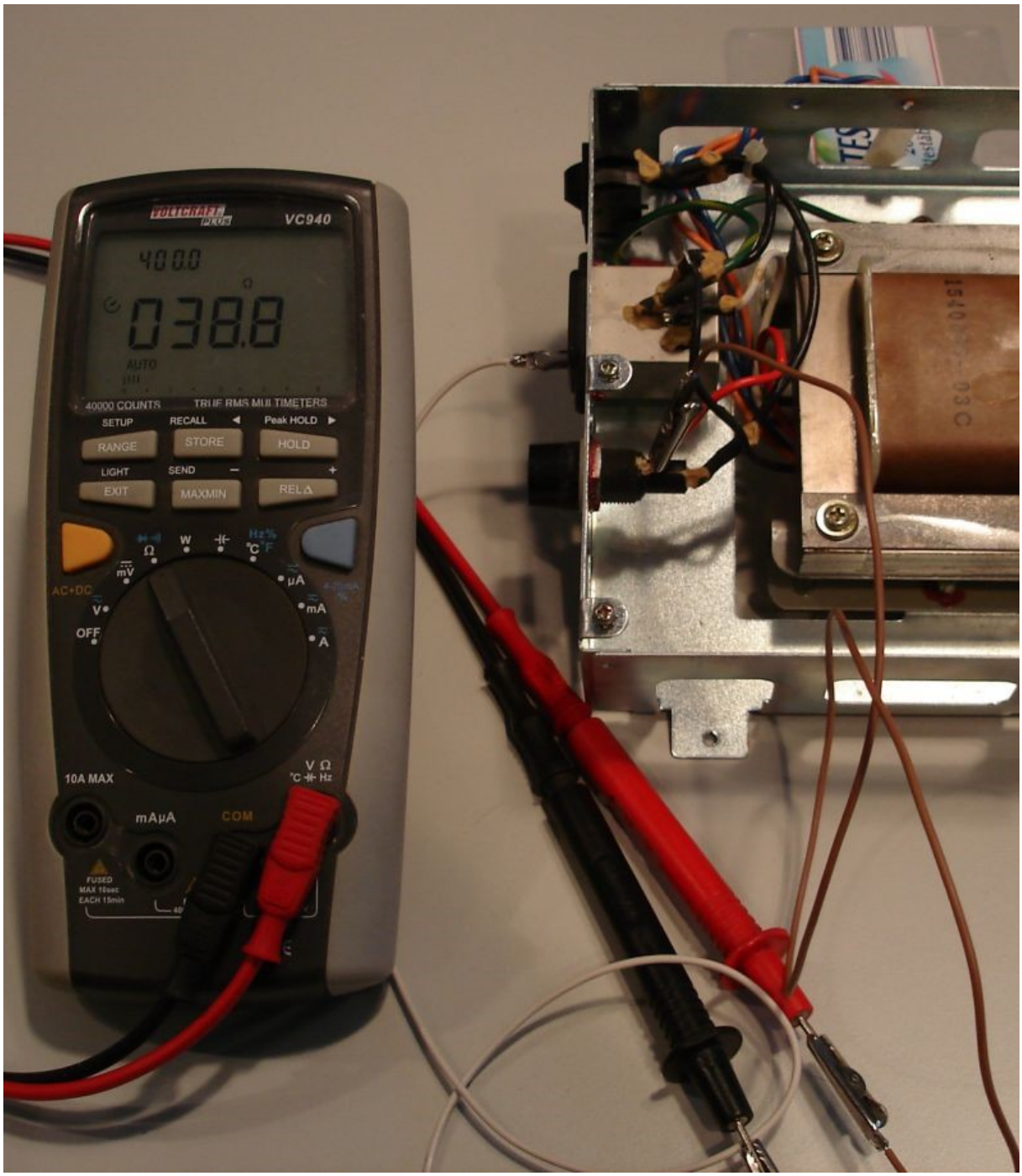
Umbau auf 240 Volt

!!!ACHTUNG!!! Netzspannung, Dieser Teil ist nur für Leute die sich damit auskennen und an Netzspannung arbeiten dürfen.

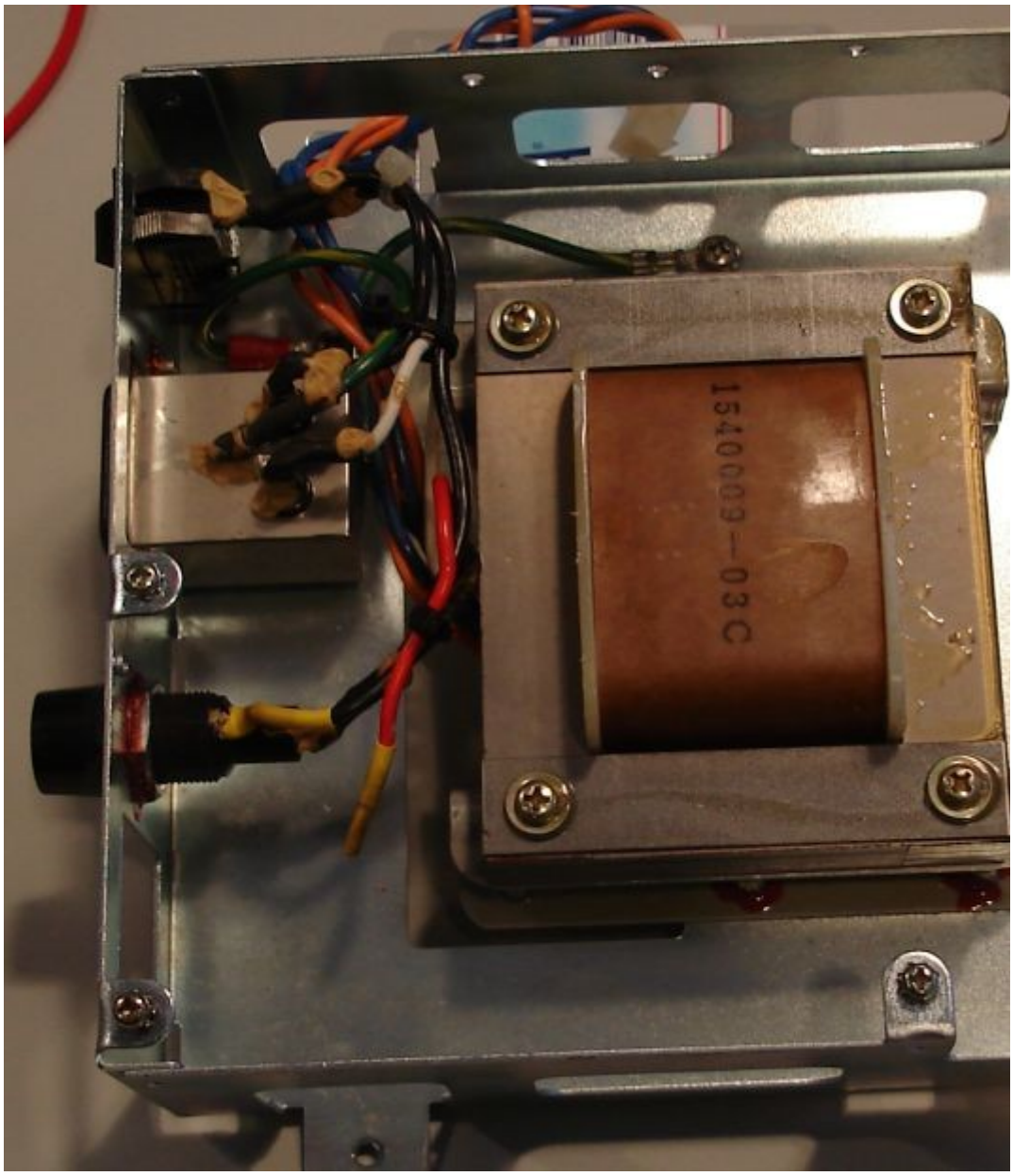
Die 1541 ist ab Werk mit einem Transformator für die Netzspannungen 220V und 240V ausgerüstet. Ausgeliefert wurden die Geräte immer mit der Einstellung 220V. Heutzutage beträgt die Netzspannung 230V und somit wird im Transformator und der Regelelektronik ein größerer Anteil der aufgenommenen Leistung in Wärme umgesetzt. Mehr Wärme stresst die Bauteile der 1541 und der thermische Verzug der Mechanik im Betrieb ist höher. Der Transformator kann sehr leicht auf 240V umgebaut werden, was bedeutet, dass die erzeugte Abwärme um ca. 30% sinkt.



Auf dem Bild ist zu erkennen das der Transformator primär seitig 3 Leitungen hat, Weiß für den Nullleiter, rot für 220V und schwarz für 240V. Die schwarze Leitung hängt normalerweise isoliert am Kabelbaum. Man kann sich nicht darauf verlassen das sämtliche Transformatoren die gleiche Belegung und Leitungsfarben haben. Deshalb ist es Sinnvoll das nachzumessen.

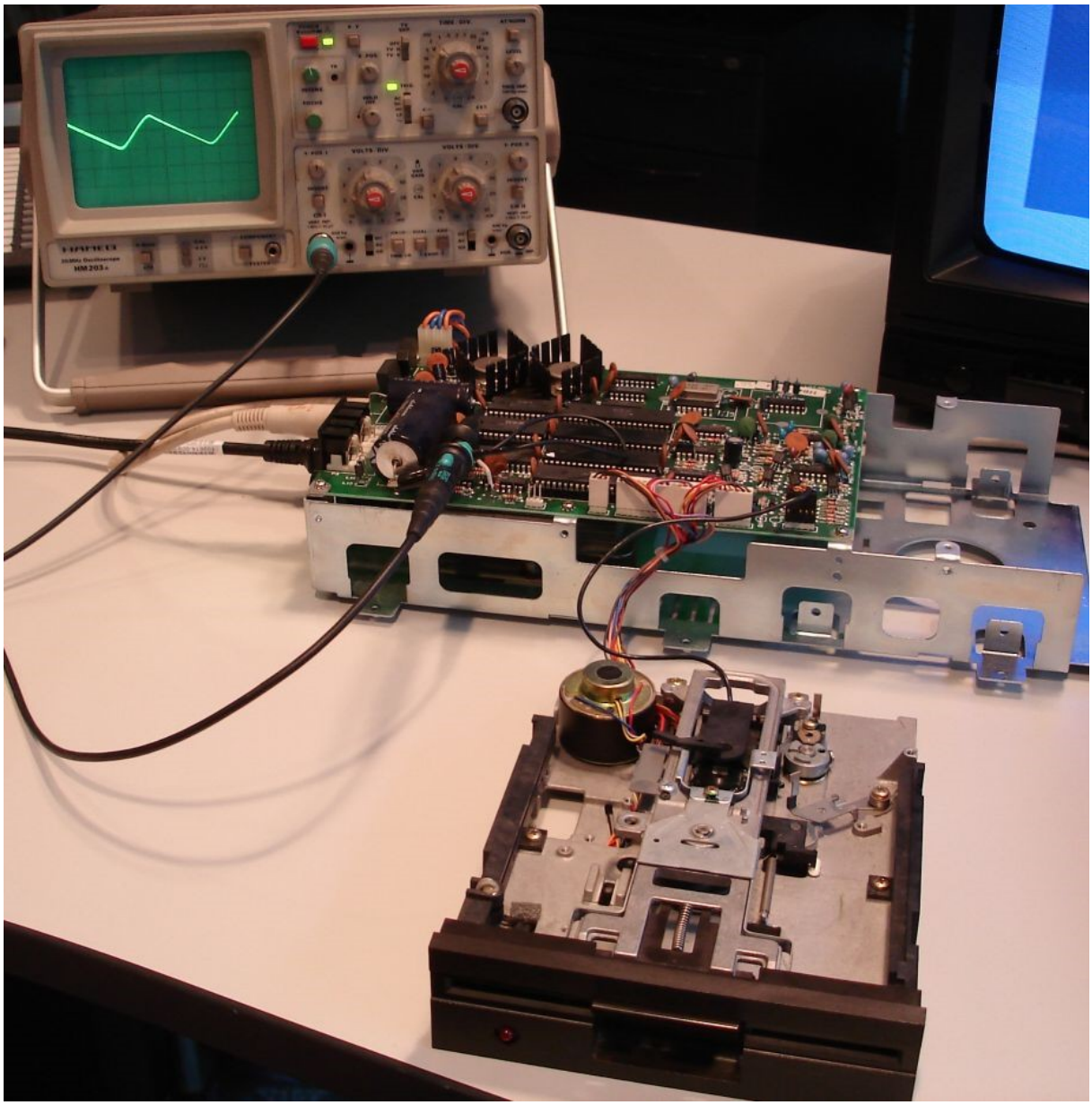


Der Schrumpfschlauch der roten Leitung wird am Sicherungshalter entfernt sowie die schwarze freihängende Leitung ab isoliert. Dann wird mit dem Multimeter im Widerstandsmessbereich von der weißen Leitung zur roten Leitung gemessen. Ein Widerstand von ca. 38 Ohm zeigt sich wenn der Transformator für 220V-Betrieb eingestellt ist. In diesem Fall muss diese und die schwarze Leitung gegeneinander getauscht werden. Eine Messung zwischen der weißen und der schwarzen Leitung sollte ca. 42 Ohm ergeben und ist die Einstellung für 240V.

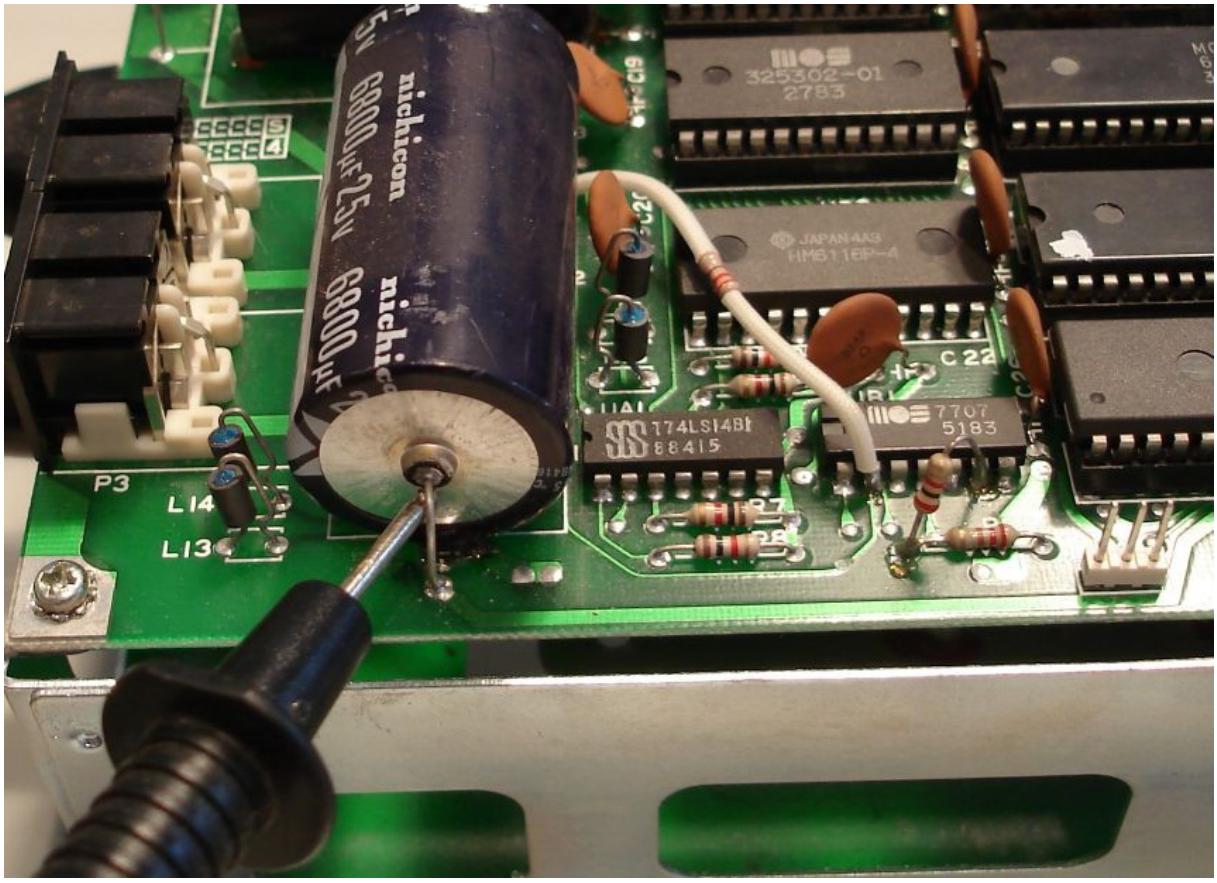


Hier ist zu sehen dass die rote und die schwarze Leitung gegeneinander getauscht worden sind und alle netzspannungsführenden Teile wieder mit Schrumpfschlauch isoliert worden sind.
Der Umbau auf 240V ist damit abgeschlossen.

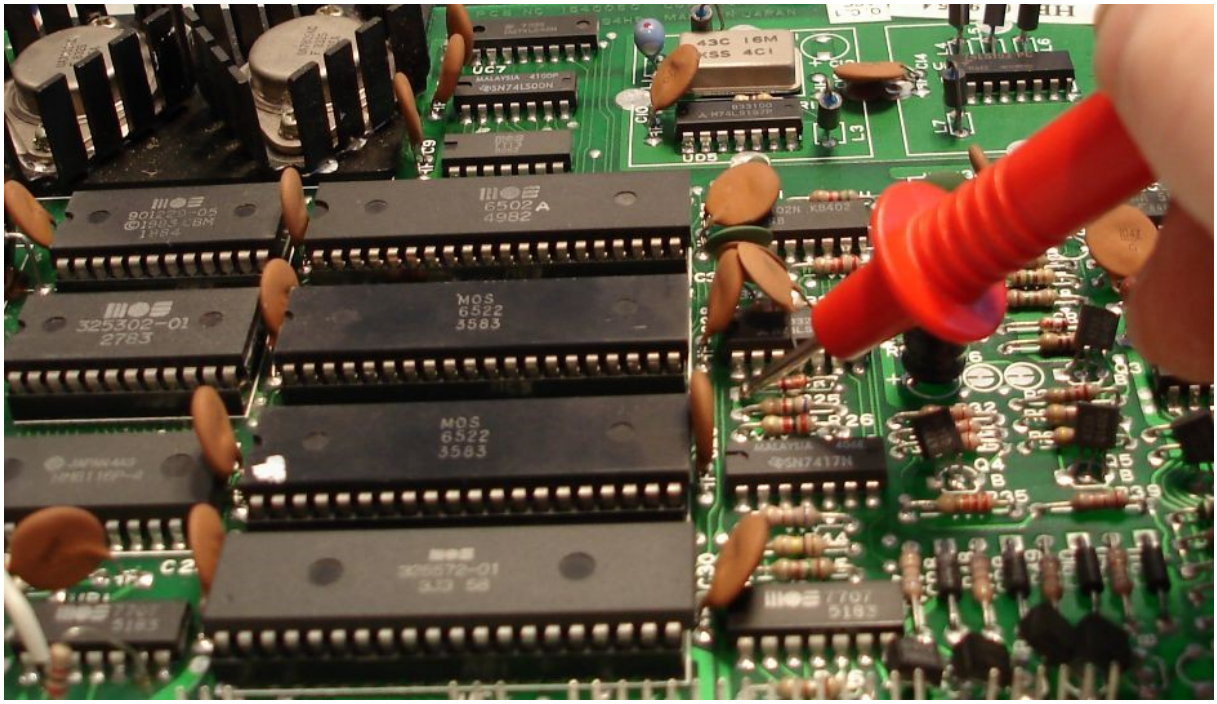
Überprüfung der Spannungsversorgung



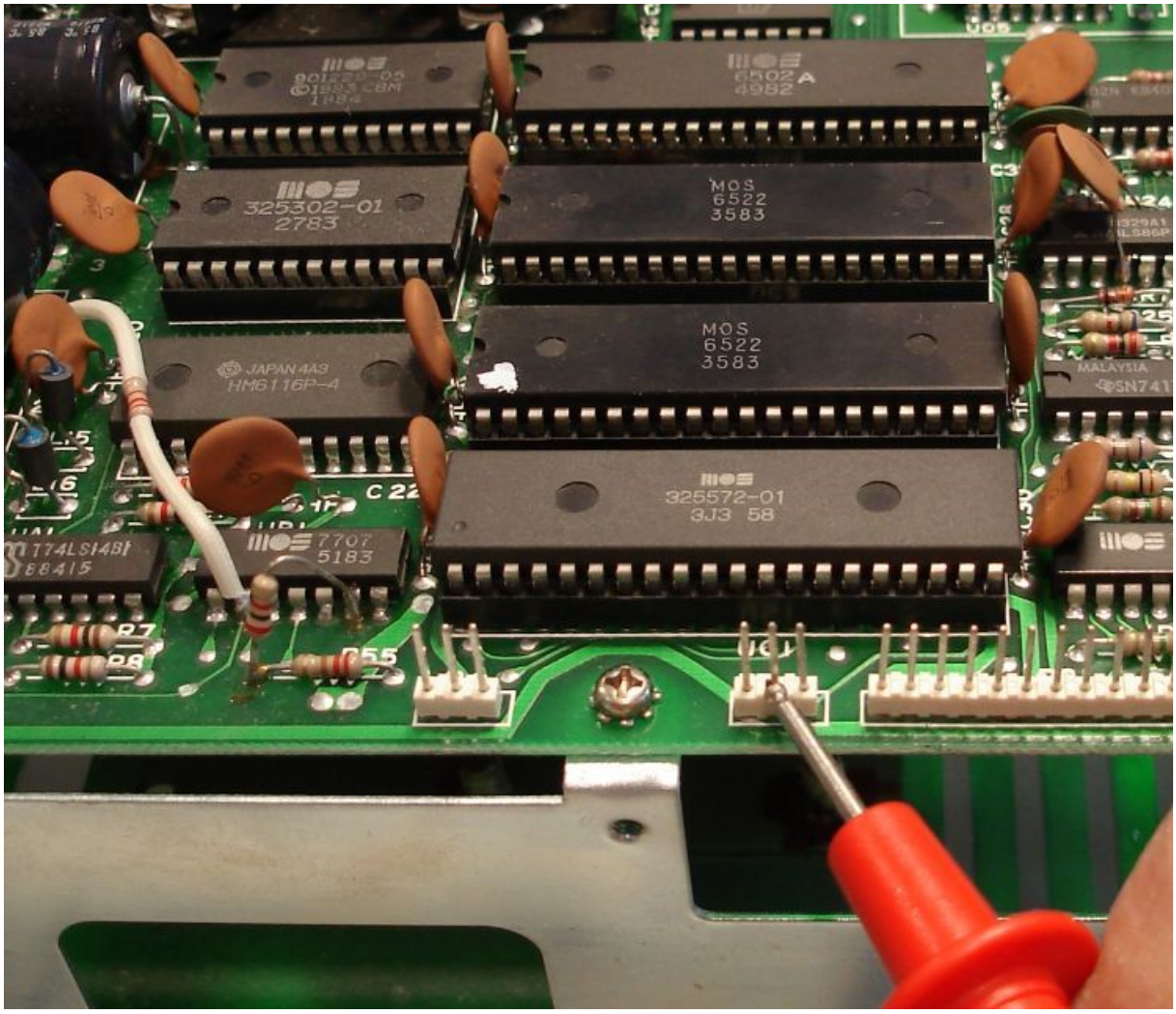
Die Platine wird zunächst wieder mit dem Chassis verschraubt und das Laufwerk davorgelegt und angeschlossen. Mit dem Multimeter wird jetzt überprüft ob die Betriebsspannungen von 5V und 12V in Ordnung sind.



Als Massebezug für alle folgenden Messungen ist der Masseanschluss des Elektrolytkondensators C17 zu nehmen.



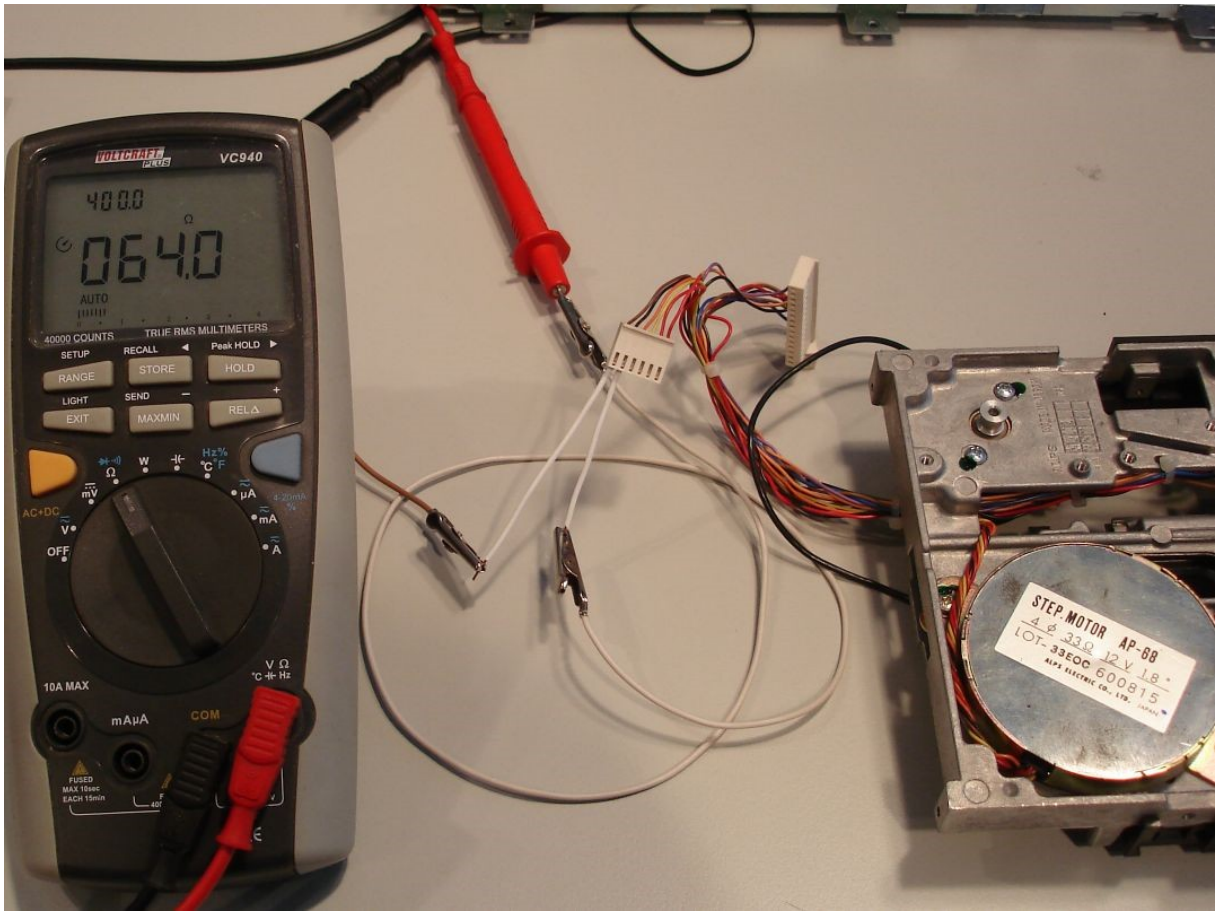
Die 5V werden wie im Bild gezeigt an der Diode CR7 gemessen. Eine Spannung zwischen 4,8 und 5,2 Volt ist in Ordnung.



Die 12V werden am Motorstecker mittlerer Pin gemessen. Eine Spannung zwischen 11,5 und 12,4 Volt ist in Ordnung.

Elektrische Überprüfung der Mechanik

Der Kopf und der Steppermotor sind mit dem Multimeter per Widerstandsmessung prüfbar. Alle gemessen Werte können so um 2-3 Ohm von den hier dargestellten Werten abweichen.



0,6mm Drahtstücke passen exakt in den Stecker und erleichtern die Messung. Der Steppermotor hat 4 Wicklungen von denen jeweils 2 in Reihe geschaltet sind. Im Mittelpunkt der Reihenschaltungen ist jeweils ein Abgriff für die Spannungsversorgung (Rote Leitungen).

Widerstandswerte nach Leitungsfarben:

Braun nach Schwarz: 64 Ohm
Braun nach Rot: 32 Ohm
Schwarz nach Rot: 32 Ohm

Orange nach Gelb: 64 Ohm
Orange nach Rot: 32 Ohm
Gelb nach Rot: 32 Ohm

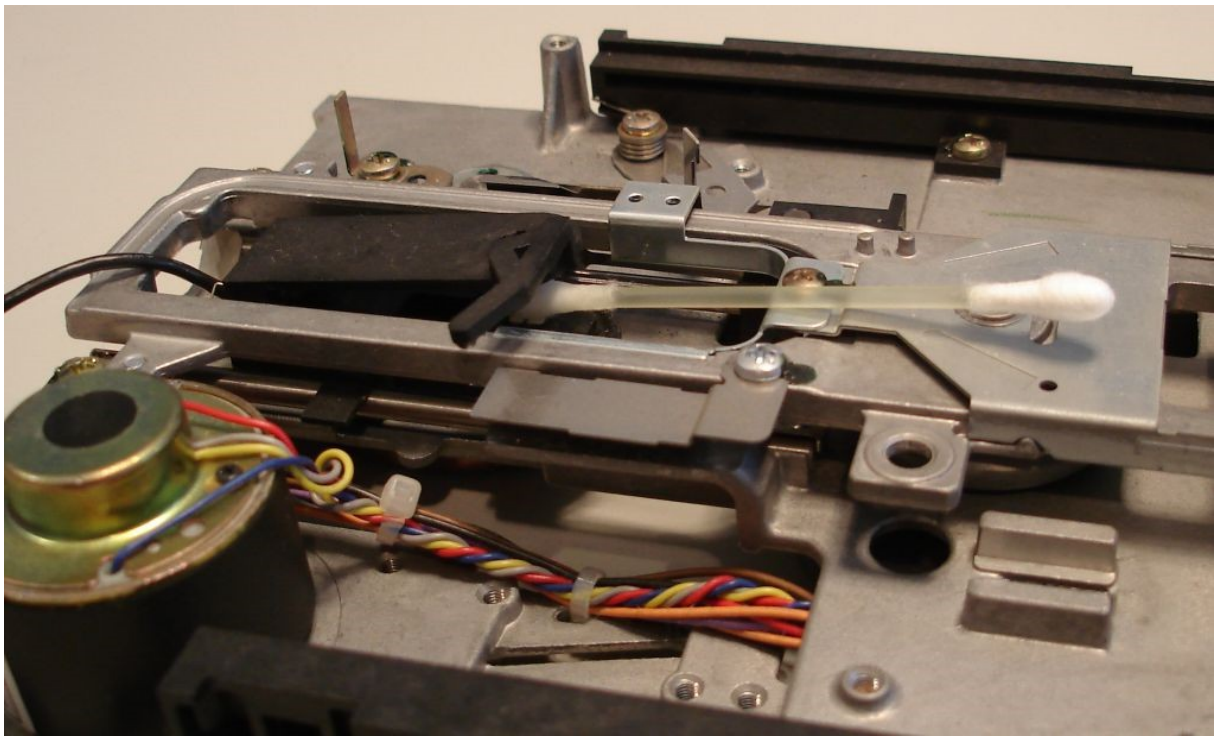
Rot nach Rot: offen (unendlich hoher Widerstand)

Kopf durchmessen:

Rot nach Blau: 32,5 Ohm
Rot nach Gelb: 27 Ohm
Rot nach Weiß: 16,5 Ohm
Gelb nach Weiß: 10,5 Ohm
Gelb nach Blau: 26,5 Ohm
Weiß nach Blau: 16,5

Reinigung und Schmierung

Der Kopf setzt sich mit der Zeit mit Schmutz und Diskettenabrieb zu und die Gleitschienen ziehen Staub an, so dass die Schmierung langsam schlechter wird.



Der Kopf wird mit Isopropylalkohol gereinigt, dazu ein Q-Tip mit Alkohol tränken, zwischen Andruckfilz und Kopf einklemmen und 2 Minuten einziehen lassen. Danach den Kopf und Filz sauberrubbeln und mit der trockenen Seite des Q-Tip den Kopf nachpolieren. Danach das Laufwerk mit offener Klappe 15 Minuten stehen lassen. Den Andruckfilz nur soweit hochklappen wie es zu Reinigung unbedingt nötig ist, sonst leiert die Andruckfeder aus.

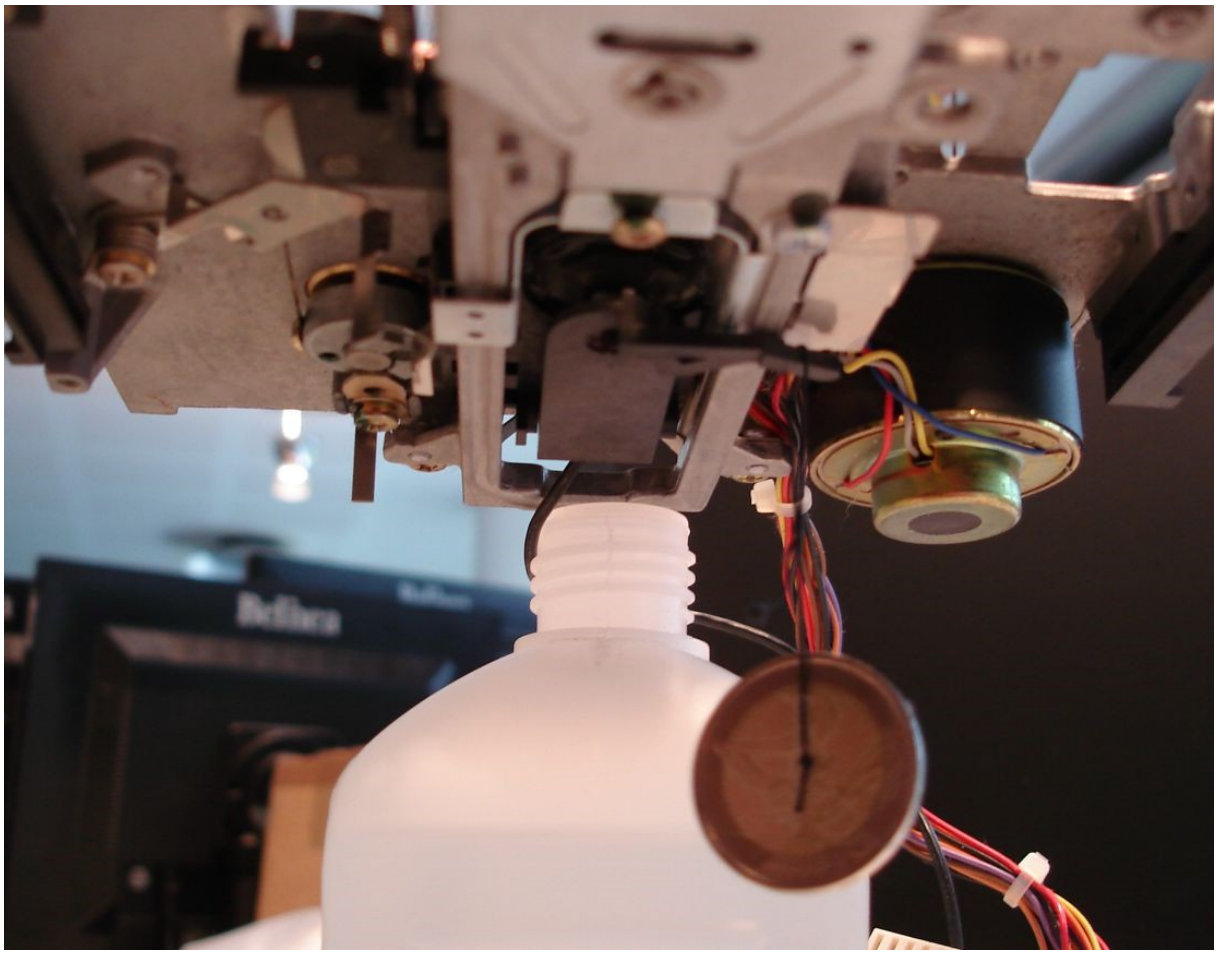
In der Zwischenzeit kann der Riemen gereinigt werden. Dazu den Riemen abnehmen und 2-3-mal durch ein alkoholgetränktes Tuch ziehen.

Zum Schluss werden die Gleitschienen für den Kopf leicht eingeölt. Etwas Öl auf einen Q-Tip auftragen und damit die Schienen einreiben. Es wird NICHTS anderes geölt!

Überprüfung des Anpressdrucks

Die Feder in der Kopfeinheit sorgt dafür dass bei geschlossener Laufwerksklappe die Diskette mit einem definierten Druck auf den Kopf gepresst wird. Diese Feder kann ausleiern (speziell wenn der Andruckfilz zu weit hochgeklappt wird. Die Gerätschaft um den richtigen Druck zu ermitteln hat wahrscheinlich niemand, daher gibt es eine simple Lösung.

Eine 2-Euro Münze mit einer Garnschlaufe als Prüfgewicht.



Das Laufwerk wird kopfüber gehalten und das Gewicht vorsichtig wie im Bild zu sehen angehängt. Eine gute Feder wird die Münze tragen können ohne dass der Andruckfilz sich nach unten bewegt. Wenn sich der Filz nur um ein paar Millimeter nach unten bewegt ist das auch noch OK, nur sehr viel mehr sollte es nicht sein.

Spurlage, Spur-0 und Drehzahl Einstellung

Spurlage:

Um die Spurlage genau einstellen zu können ist ein Oszilloskop und eine Referenzdiskette zwingend erforderlich. Wer kein Oszilloskop verfügbar hat sollte davon absehen an der Spurlage zu justieren. Anstelle der Referenzdiskette kann auch die 1541 Test/Demo Disk genommen werden, diese wurde auf einer korrekt eingestellten Maschine beschrieben.

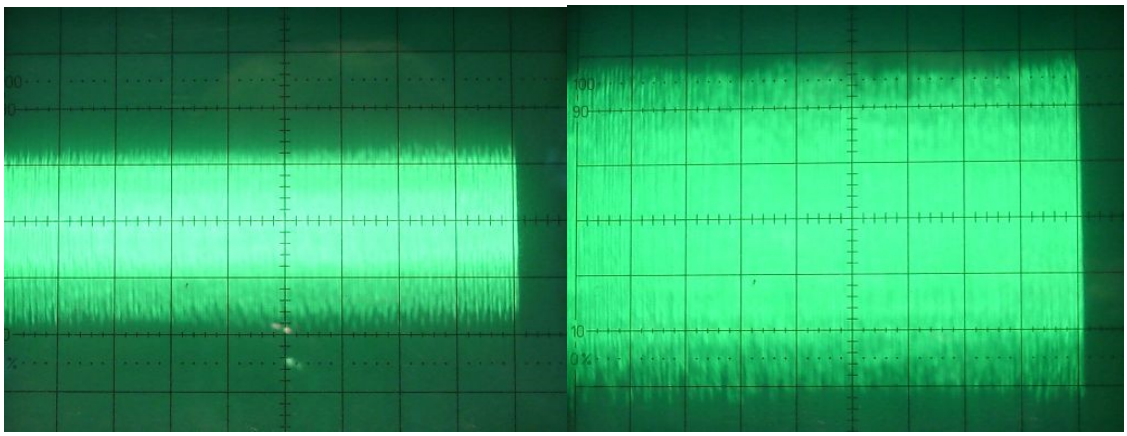
Die Messung muss mit einem 1:10 Tastkopf erfolgen, der Tastkopf wird an einem der beiden Beine der Spule L11 auf der Platine angeschlossen und die Masseklemme an Masse von C17. Das Oszilloskop wird auf AC-Messung mit Spannung 5mV/Div. und Zeitbasis 200µs/Div. gestellt.

Die beiden Schrauben die den Steppermotor festhalten werden so weit gelöst das sich der Motor gerade eben noch nicht von selbst verstellen kann. Den C64 mit der Platine verbinden Das Laufwerk seitlich hinstellen und die Floppy einschalten und die Test/Demo Diskette einlegen. Auf dem C64 wird ein kleines Basicprogramm erstellt das nichts anderes tut als ständig Spur 18 zu lesen:

10 LOAD"$",8,1

Dieses Programm starten und dann die Wellenform auf dem Oszilloskop beobachten. Eine gut eingestellte Spurlage erzeugt Flusswechsel mit hoher Amplitude auf dem Oszilloskopbildschirm.

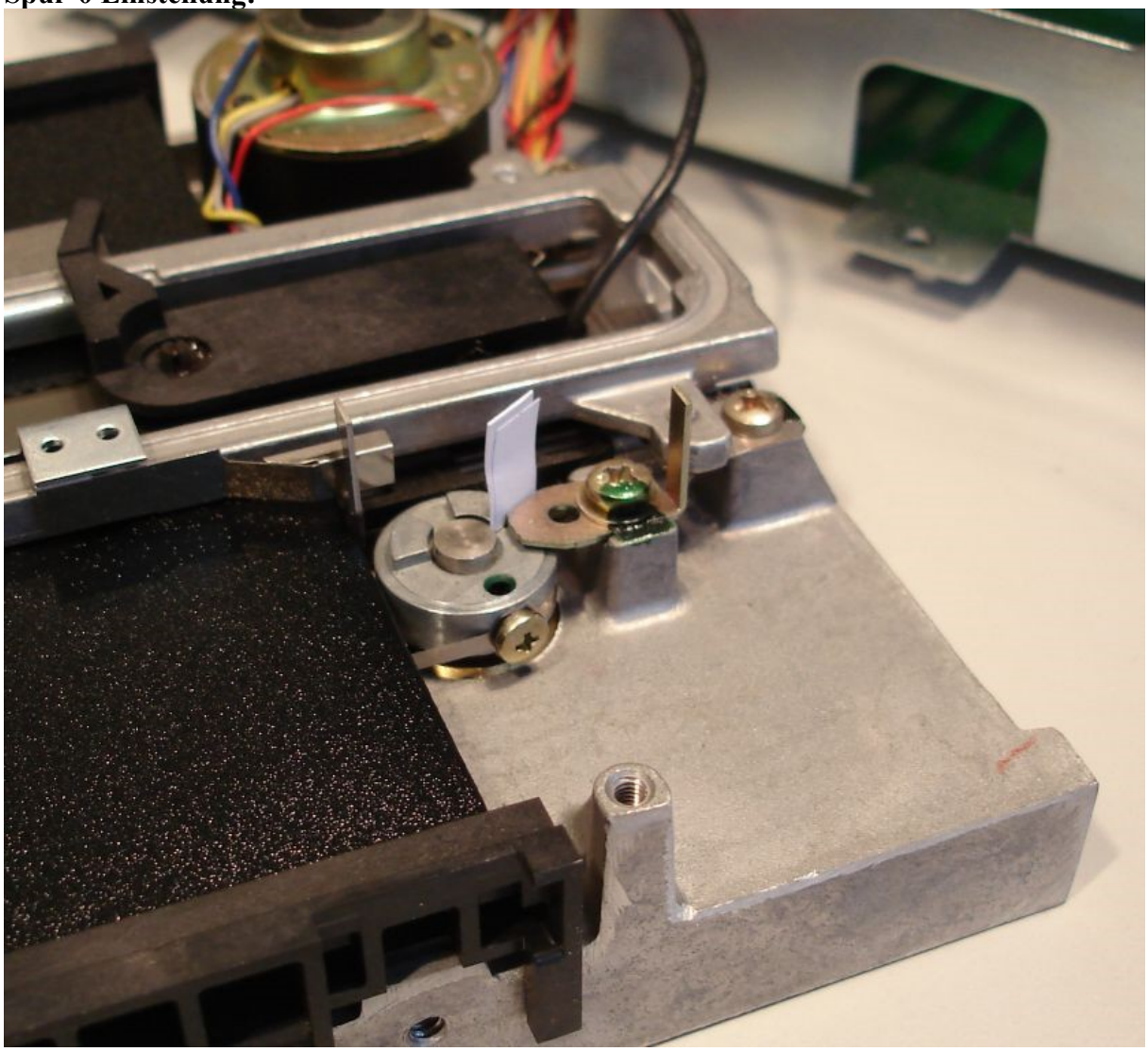
Zum Vergleich zwei Bilder:



schlechte Spurlage **gute Spurlage**

Die Spurlage kann durch vorsichtiges Drehen des gesamten Steppermotors auf maximale Amplitude eingestellt werden. Es sind wahrscheinlich mehrere Versuche notwendig weil beim Festziehen der Befestigungsschrauben der Motor wieder leicht verstellt wird.

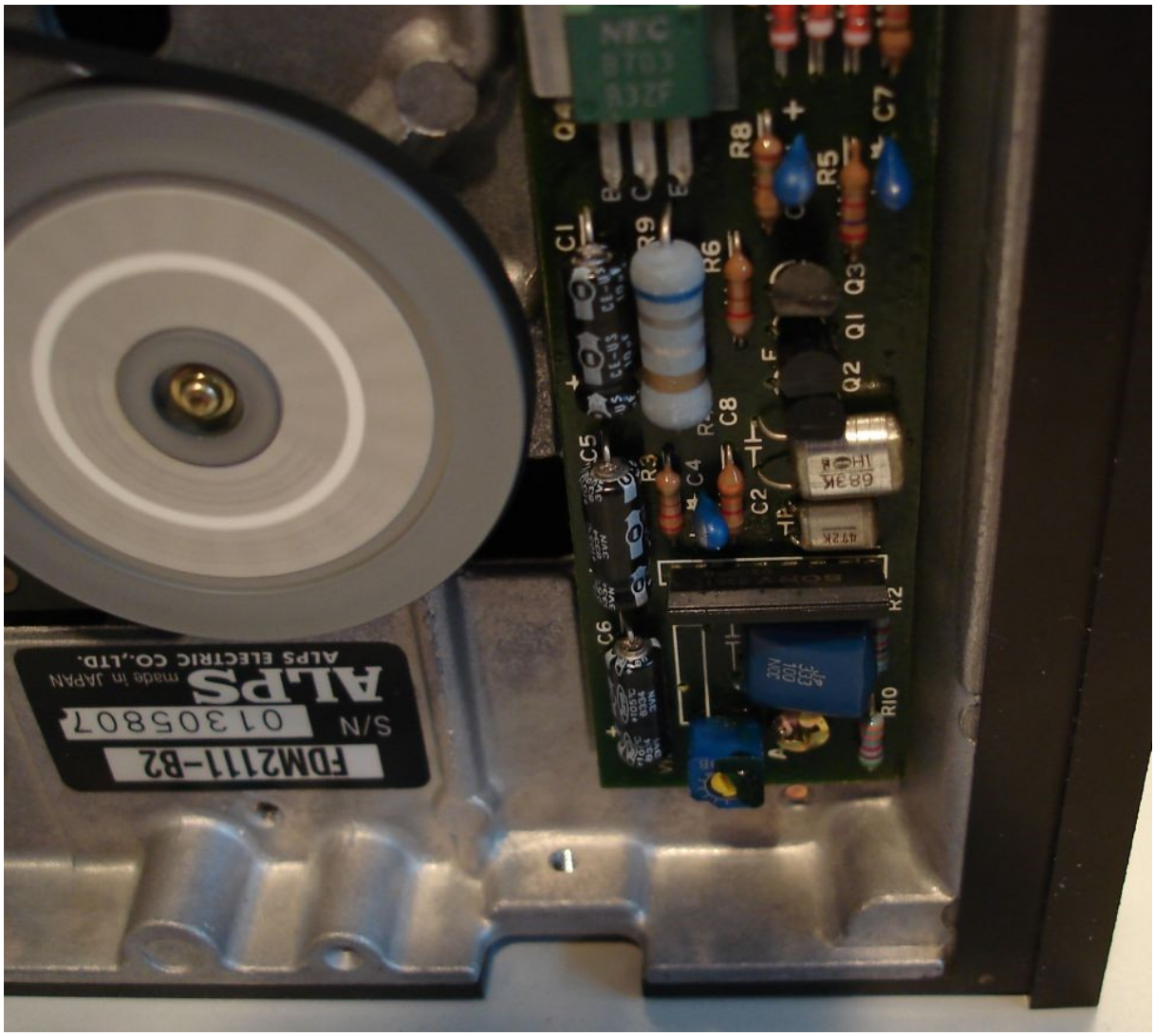
Spur-0 Einstellung:



Dazu wird die Software "Floppy Adjust" geladen und dann per Feineinstellung als Starttrack sowie Endtrack Spur 1 gewählt. Als Hilfsmittel wird ein Blatt Kopierpapier in drei kleine Streifen geschnitten und diese drei Streifen übereinandergelegt. Dieses muss dann genau ohne Spiel zwischen den Anschlagsnocken auf dem Steppermotor und dem Anschlagblech passen. Gegebenenfalls dass Blech entsprechend einstellen.

Drehzahleinstellung:

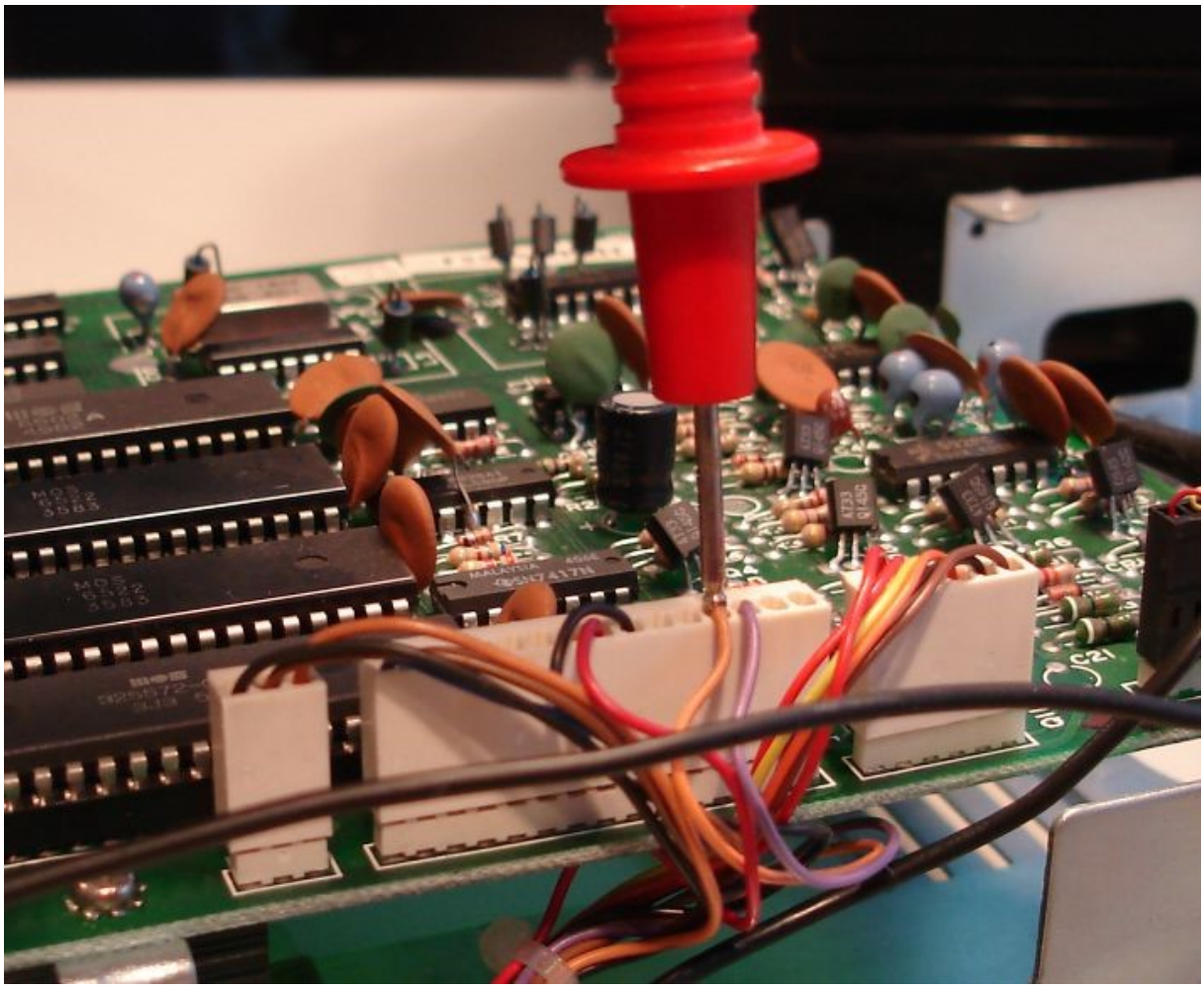
Dazu wird die Software "Kwik Copy" geladen dann wieder die Test/Demo Disk eingelegt und der Menüpunkt Drive-Speed gewählt.



Auf der Motorplatine befindet sich ein kleiner Trimmer zur Einstellung der Drehzahl. Dieser wird jetzt so eingestellt das Drive-Speed genau 310 RPM anzeigt. 310 RPM deswegen weil Kwik Copy ein Programm ist das ursprünglich für NTSC-C64 entwickelt wurde und der Timingunterschied zum PAL-C64 etwa 10 RPM beträgt.

Lichtschrankentest:

Die Lichtschranke ist erforderlich damit die Floppy einen Diskwechsel und einen Schreibschutz erkennen kann. Die Prüfung ist sehr einfach, und wird mit dem Multimeter im Spannungsmessbereich gemacht.



Die Spannung an der orangenen Leitung beträgt ohne Disk im Laufwerk ca. 5V, wenn eine Disk halb eingelegt wird muss die Spannung auf ca. 0,15V fallen.

Zum Schluss wird noch das Öl auf den Gleitschienen etwas besser verteilt, indem noch mal das Programm "Floppy Adjust" gestartet wird und im Menü Feineinstellung alle Vorgaben StartTrack, Endtrack und Zeit so belassen werden. Das Programm sollte ca. 2 Minuten laufen, der Kopf bewegt sich dabei ständig von Track 1 bis 35 verteilt dabei das Schmiermittel gleichmäßig.

Endtest:

Die 1541 wird wieder komplett zusammengebaut und das Programm "PERFORMANCE TEST" von der Test/Demo Disk gestartet. Wenn sich hier keine Probleme zeigen dann war die Wartung erfolgreich.

MMC2IEC – SD2IEC: SD Card am C64

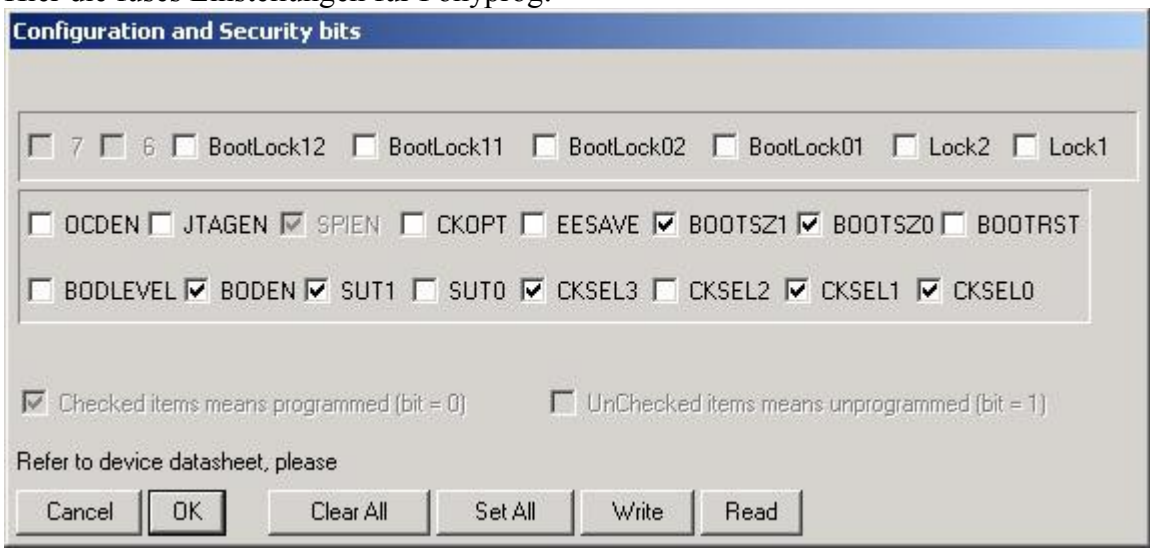
Historie / Hintergründe

Statt einer Floppy 1541, welche doch auch irgendwann mal das Zeitliche segnen wird, könnte man auch ein etwas moderneres und dabei deutlich kleineres Speichermedium verwenden. Aus dieser Idee entstand die 1541-III von Jan Derogee, welche mit einem PIC-Microcontroller arbeitet. Pyrofer hatte bereits mit dem MMieC Projekt etwas ähnliches geschaffen. Als dann der C64DTV auf den Markt kam, hat sich Pyrofer zusammen mit Jussi Saily daran gemacht, die 1541-III und das MMieC umzubauen, damit das Resultat am C64DTV betrieben werden kann. Daraus entstand die 1541-III DTV - immer noch mit einem PIC. Lars Pontoppidan hat dann die 1541-III DTV für seinen C64DTV Mod auf einen AVR ATMega32 umgesetzt. So entstand sein MMC2IEC device. Er hat dann auch noch eine Version erstellt, welche wiederum an den C64 angeschlossen werden konnte - damit hat sich der Kreis zur ursprünglichen 1541-II wieder geschlossen. Das letzte, was Lars gebaut hat ist ein MMC2IEC, welches sowohl am C64DTV als auch am C64 arbeitet - diese hat er auch zum Verkauf angeboten. Basierend auf den Schematics von Lars gab es dann im Forum64 einige Nachbauten für den C64DTV. Im Gegensatz zu Lars verwendete mancher dabei nicht den ATMega32 als TQFP (44pin SMD) sondern als DIP (40pin), da dieser deutlich einfacher zu handhaben ist. AndyDTV benutzte dabei die L-Variante (3.3V) für seinen Nachbau. Im Gegensatz dazu verwendete Shadowolf zwar den TQFP, änderte aber ein paar Ports. Die aktuelle Firmware SD2IEC passt schon nicht mehr in einen Atmega32, sondern der Pin-kompatible Atmega644 wird benötigt.

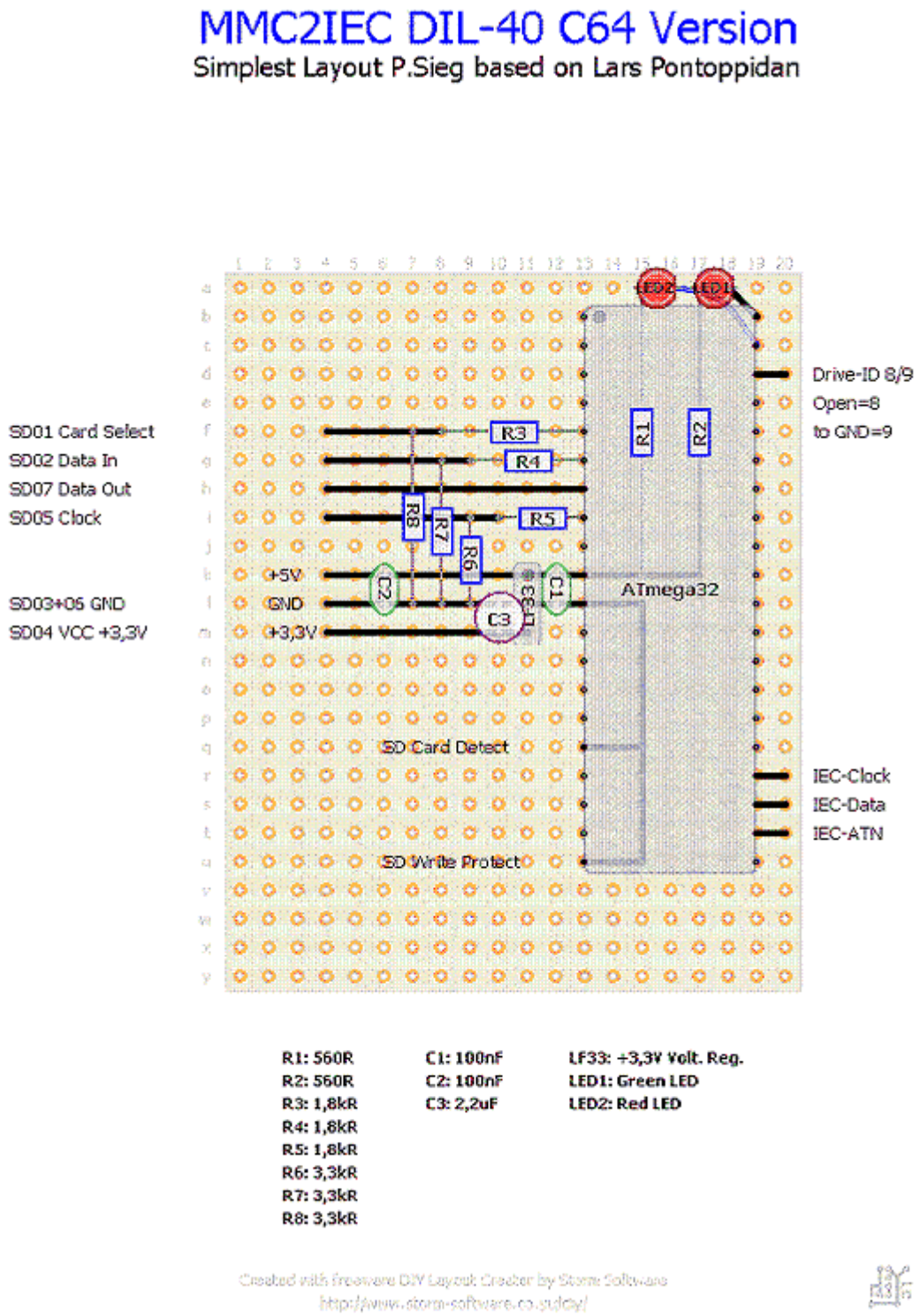
Mein Nachbau

Da mir der Nachbau mit SMD nicht so liegt, habe ich mich für einen Nachbau auf Lochraster entschieden. Programmiert habe ich mit dem Pollin Eval. Board. Hier der mmc2iec.zip HEX-File. -> http://petersieg.webng.com/mmc2iec/mmc2iec.zip

Hier die fuses Einstellungen für Ponyprog:

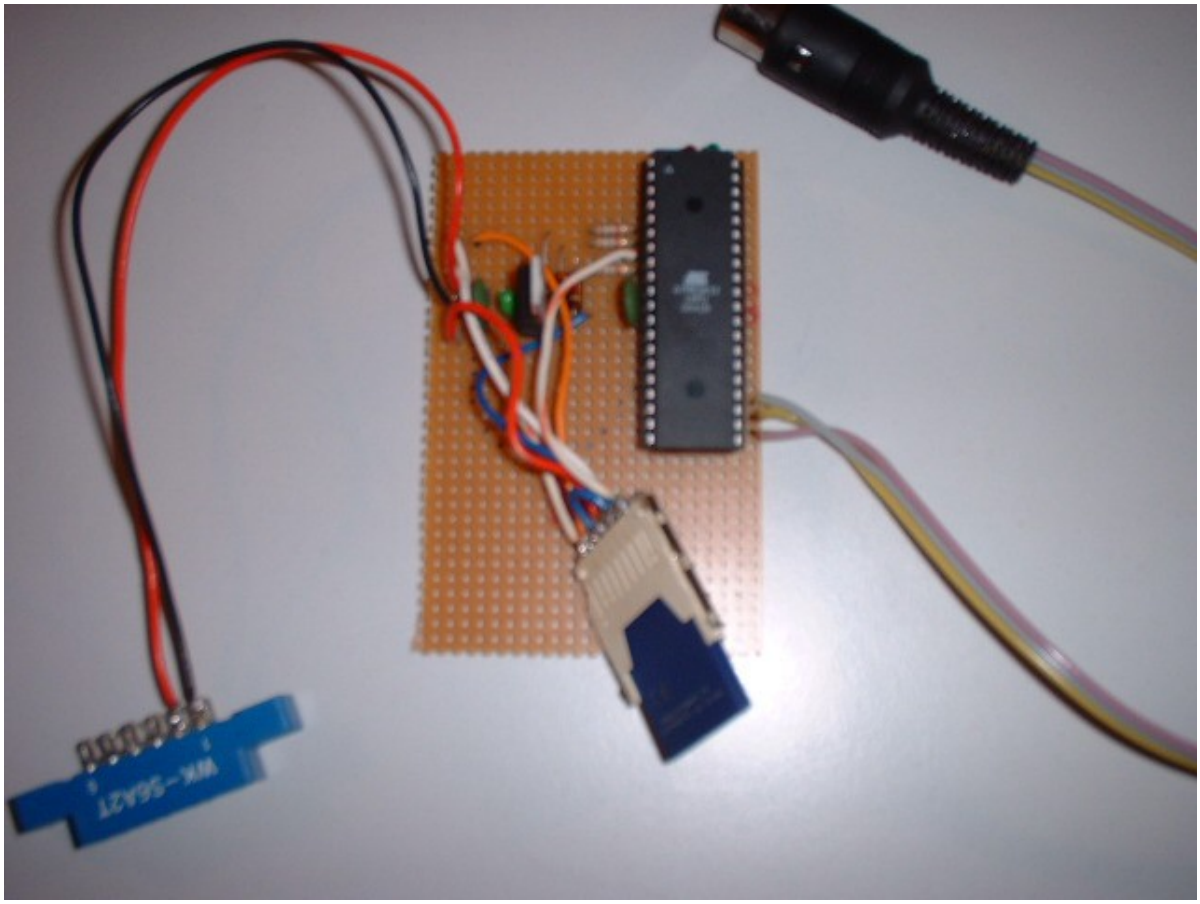


Lochrasterschaltplan



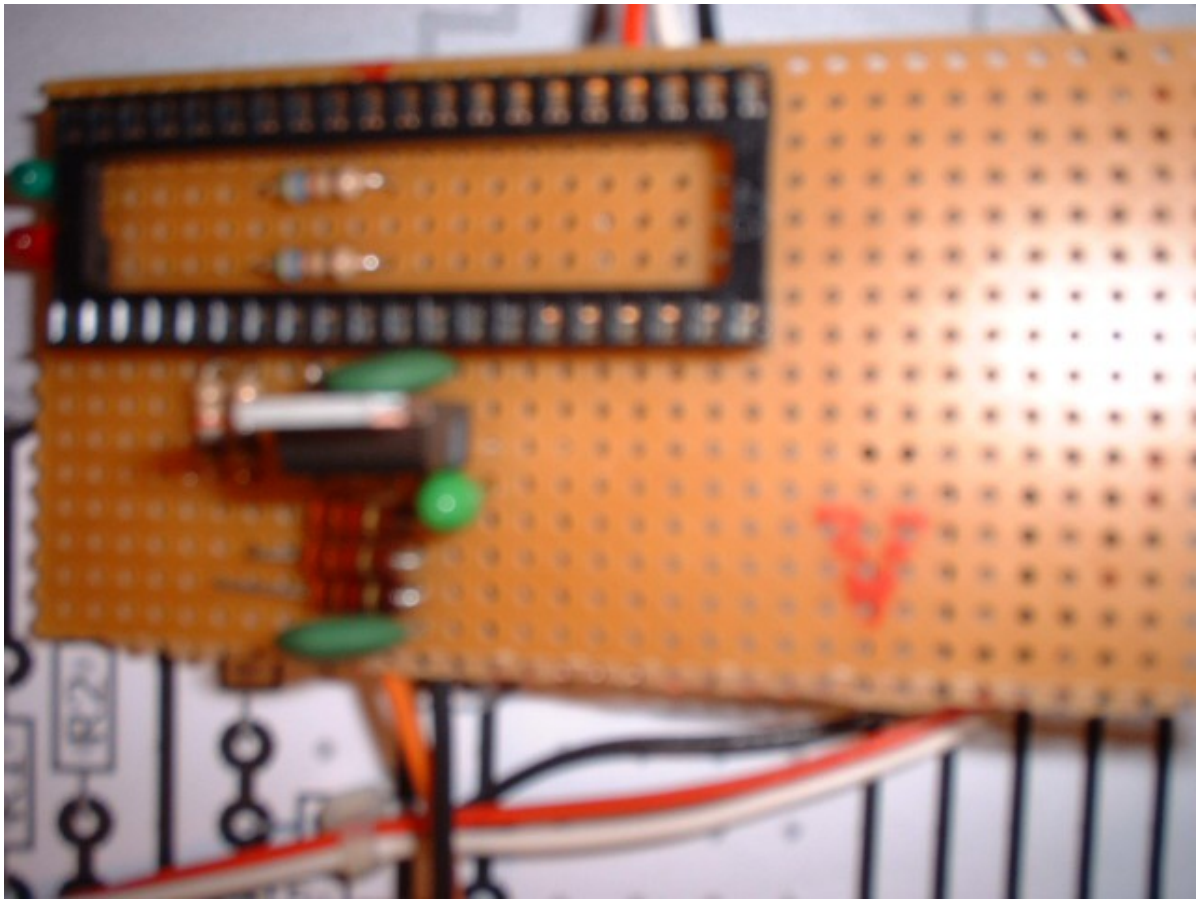
Anstatt dem LF33 3,3V Regler kann man auch z.B LP2950 3V Regler verwenden.. Der LF33 im TO-220 Gehäuse ist halt ziemlich groß. Auch 2 Si-Dioden in Reihe zur Spannungsreduktion funktionieren.

Hier ein Photo des fertigen Aufbaus (SD-Slot wird noch mit Heißkleber fixiert) Wie man sieht, habe ich die 3 IEC Leitungen direkt an der rechten Seite des ATmega32 angeschlossen. Die beiden 560 Ohm Widerstände zu den LED's sind unterhalb des ATmega im Sockel 'verstaut':



Hier ein weiterer Aufbau in einen USB Multicard Reader (Reichelt 4,80€).
Usb Reader 'zerlegt' und SD-Slot abgelötet:

Aufbau auf Lochraster. Noch ohne ATmega32. Man sieht die beiden im Sockel liegenden 560 Ohm Widerstände für die LED's:



Hier im Gehäuseboden eingepasst:

Hier im Betrieb. (In den Deckel mußte ein Loch gedrehmelt werden für den 3,3V Regler im TO-220 Gehäuse ;-)):

Und hier der Größenvergleich MMC2IEC Gerät auf einer 1541C:

Bildschirmmeldungen

Tips und Hinweise

Ich habe SD Card Detect und SD Write Protect fest auf GND gelegt. Falls der verwendete SD Slot entsprechende Signale anbietet, kann man diese aber natürlich auch nutzen. Dann sind die PIN's entsprechend mit den Slot-Anschlüssen zu verbinden und die GND Verbindung zu trennen!

+5V kann man z.B am Tapeport entnehmen:

```
  1   2   3   4   5   6
  === |===========
  A   B   C   D   E   F
```

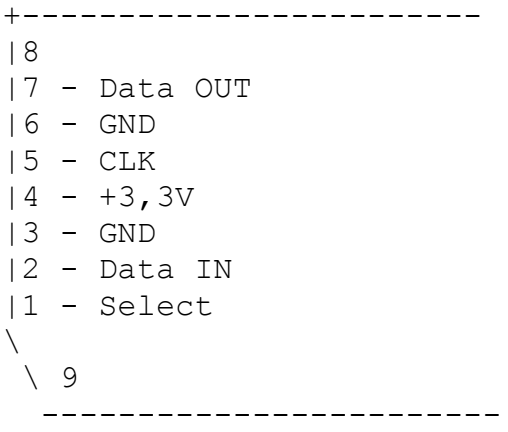
Wenn man von hinten auf den Stecker schaut.
Der untere Kontakt (A-F) ist mit dem jeweils oberen verbunden (A-1, B-2.. F-6)

+5V ist an B/2
GND ist an A/1

IEC-Buchse (in der Mitte ist Pin 6)

3 - ATN
4 - CLK
5 - DATA

SD-Card



LED
Das längere Bein ist + (Anode).

Nicht funktionierende SD Karten laufen evtl. nach dem formattieren mit diesem Tool wieder:

http://panasonic.jp/support/global/cs/sd/download/sd_formatter.html

Full Format und Adjust einstellen.

Neue Firmware sd2iec

Ingo Korb aus dem forum64 hat eine neue Firmware entwickelt! Diese ist Jiffydos kompatibel und unterstützt auch Turboload. Allerdings muß für Turboload ein Quartz und 2 dazu passende Kondensatoren bestückt werden (XTAL1+XTAL2) und die Fuses entsprechen programmiert werden (Low=0xAF; High=0xD9)! Seit Version 0.7 wird auch der Fastloader des Final Cartridge III unterstützt.

Die jeweils aktuelle Firmware stellt Ingo Korb hier ins Netz:
http://www.sd2iec.de/
Darauf achten, das die *lp für die LarsP Pinbelegung verwendet wird!

Im Unterschied zur orig. LarsP Firmware, erfolgt der Wechsel in ein Verzeichnis/M2I/D64 so: open1,8,15,"CD:[name]":close1 und nicht über load...!

Lochrasteraufbau mit Quartz:

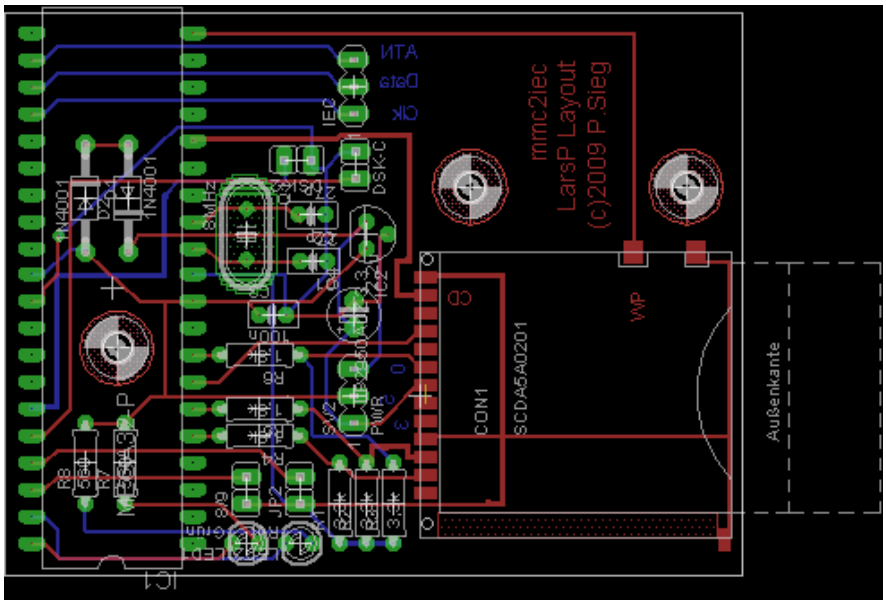
Speed-Hardcopy mit Turboload:

Eagle Layout für MMC2IEC

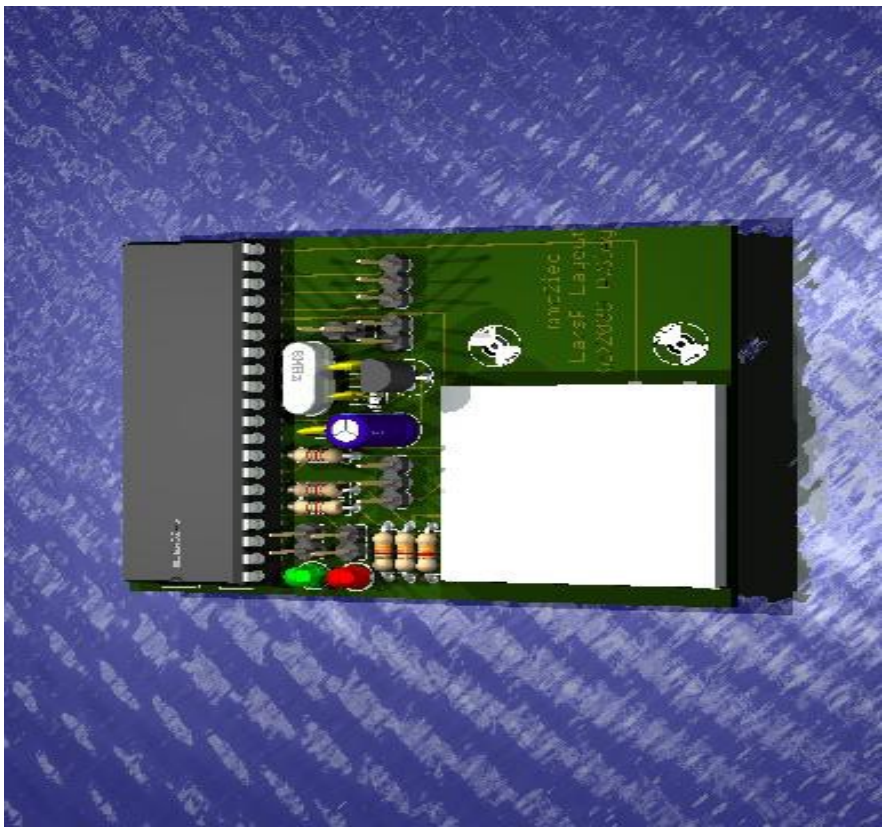
Anfang 2009 habe ich mich dann mal daran gesetzt, ein Layout in Eagle zu erstellen.

Daraus wurde dann über einen Prototypen eine kleine Sammelbestellung im forum64.

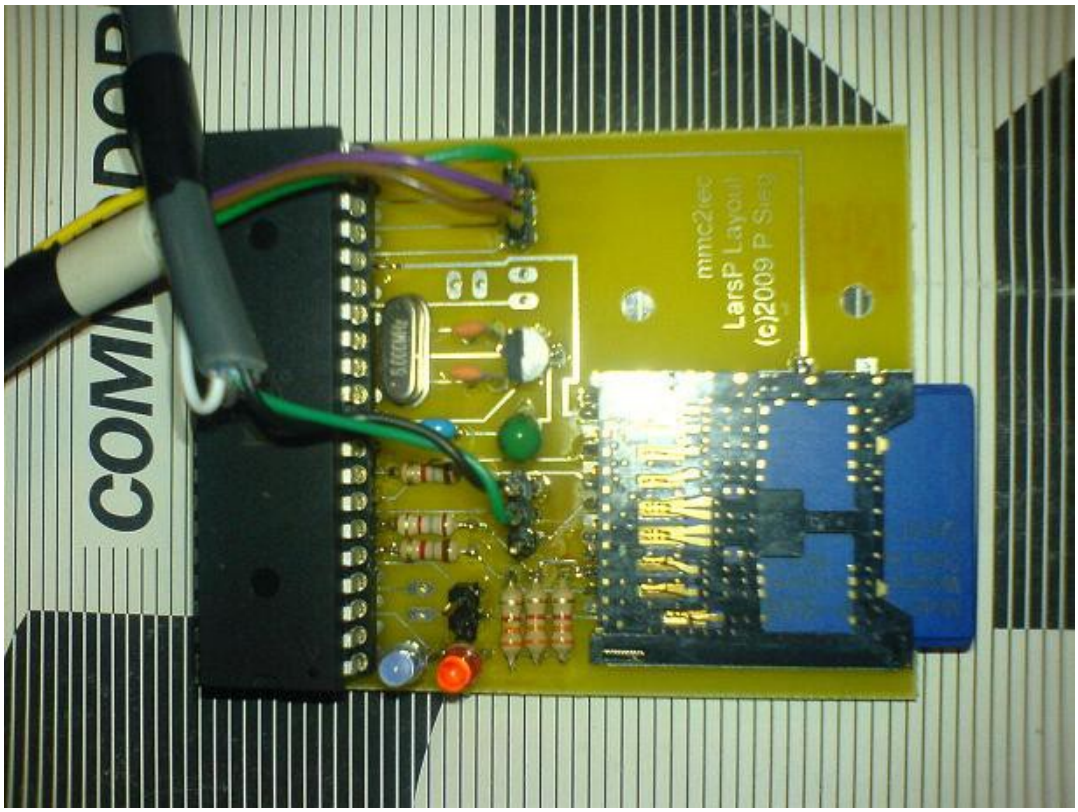
Layout:



3D gerendert mit eagle3d:



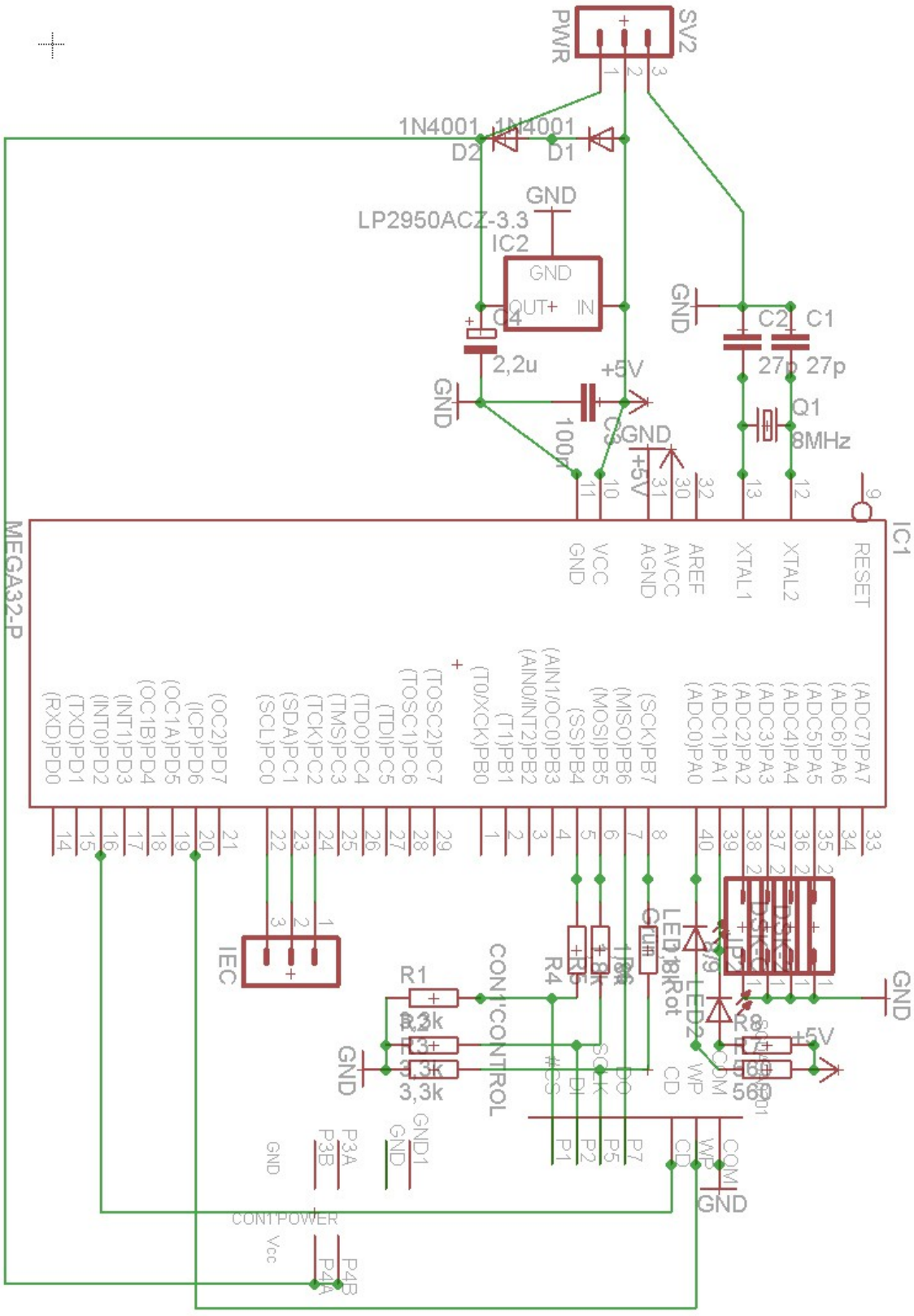
Prototyp:



Prof. Platine (www.pcbcart.com):

Schaltplan:

Wichtig: Entweder 3V Spannungsregler bestücken ODER die beiden SI-Dioden! Nicht beides! Zuerst den SD-Slot verlöten und alle Anschlußpunkte durchmessen.



IEC-ATA V2 – IDE Laufwerke am C64

Links:

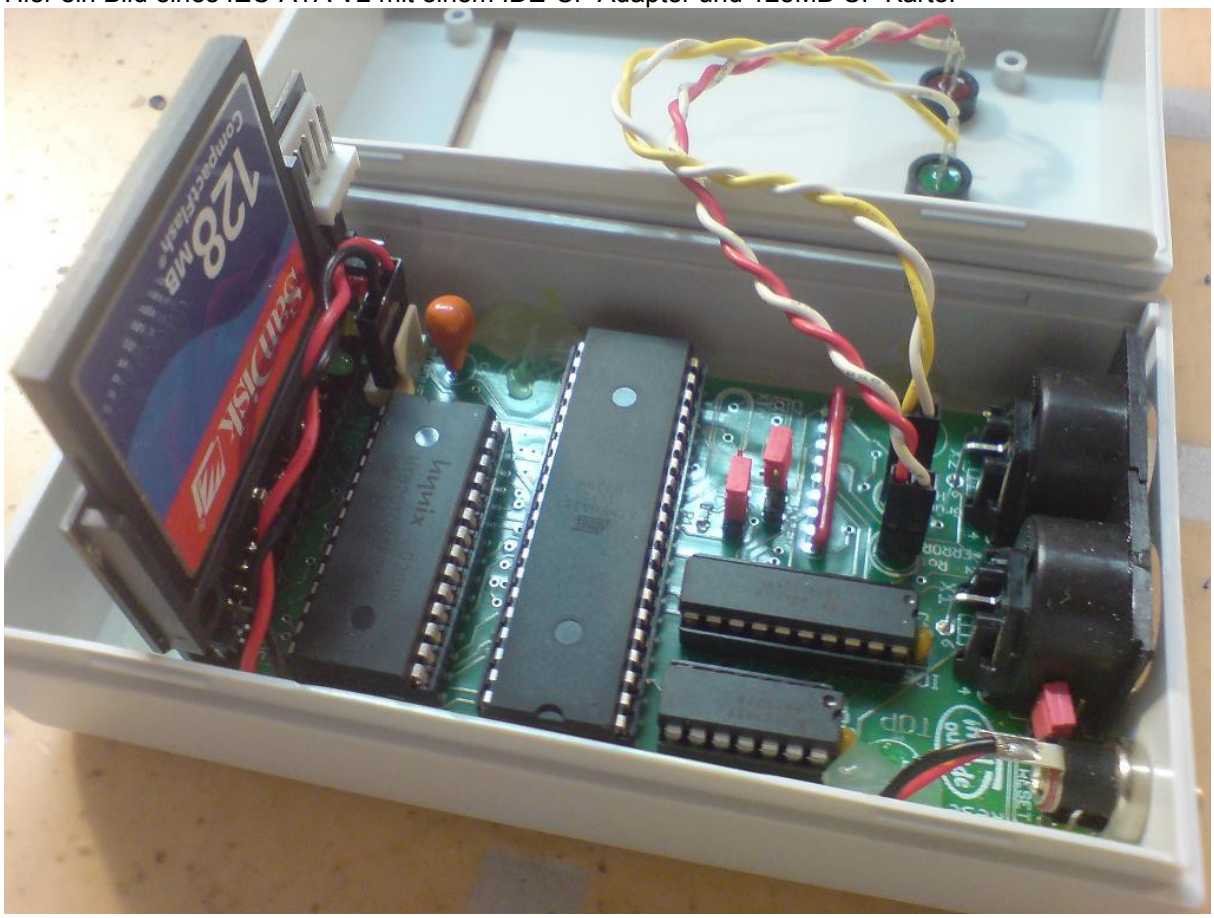
http://www.djupdal.org/cbm/iecata/
http://www.dienstagstreff.de

Das IEC-ATA beruht auf einer Entwicklung aus 2002 von Asbjørn Djupdal. Seit ca. 2004 hat sich die Fan-Gruppe des sog. Dienstagstreffs dieser Entwicklung angenommen und weiterentwickelt. Dies führte dann schließlich zur Version V2 mit einem Atmega162 und einem entsprechendem Platinenlayout. Ab ca. 2005 entwickelte Jochen Adler (NLQ) dann eine eigenen Firmware, die das FAT32 Format der IDE Massenspeicher unterstützte. Somit können Daten zwischen C64 und PC problemlos ausgetauscht werden.

Als Besonderheit im Vergleich zum MMC2IEC/SD2IEC ist es hier möglich, den Atmega162 direkt vom C64 aus mit seiner Firmware zu versehen!

Über den Seiten des Dienstagstreffs sind weitere Information auch zu anderen interressanten Projekten verfügbar. Von Zeit zu Zeit wird dort auch eine Sammelbestellung von Bausätzen organisiert.

Hier ein Bild eines IEC-ATA V2 mit einem IDE-CF Adapter und 128MB CF Karte:



Tapelaufwerk – Cassiopei

Links:
http://jderogee.tripod.com/projects/Cassiopei/Cassiopei.htm

Jan Derogee hat ein Laufwerk entwickelt was am Kassettenport arbeitet und für CBM, C64, C16, Plus 4 und VC20 funktioniert (Quasi an allen 8-bit Commodore Systemen mit Tapeport). PRG Dateien werden 50 mal schneller als vom originalen Tape geladen. Ein systemspezifisches Lade/Menueprogramm wird mit den PRG/TAP Dateien am PC auf das Gerät gespielt. Am Retro-System gibt man dann einfach LOAD ein und aktiviert den Menubutton. Und schon werden alle Programme zur Auswahl angezeigt. Kein umständliches Forward, Rewind, Positionieren eines Tapes etc.

Das Cassiopei Gerät besitzt intern 8MB Flash Speicher für PRG und TAP Dateien.
D64 Dateien – Programme die evtl. auch noch über mehrere Disketten aufgeteilt sind, eignen sich nicht für das Cassiopei.

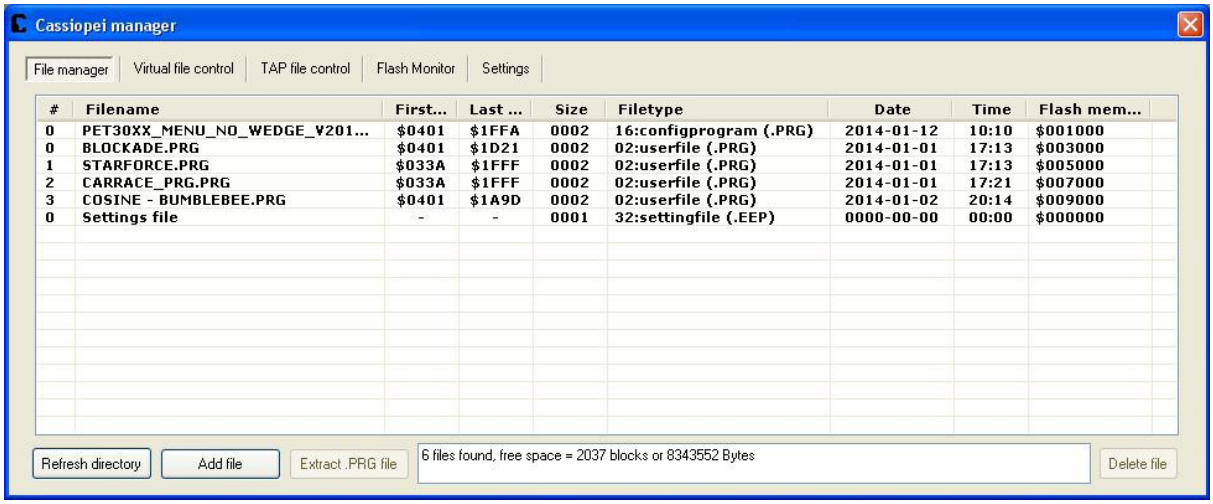
Auswahl Menue: C64

VC-20

CBM:

Cassiopei hat 3 Buttons and 1 LED, Die LED leuchtet bei Load & Save und beim bespielen vom PC. Play & Rec = Abspielen und Aufnehmen und einen Reset Button.

Cassiopei wird bespielt über eine USB Verbindung:



Dazu dient auf dem PC der Cassiopei Manager. Es können max. 254 PRG Dateien und 254 TAP Dateien aufgespielt werden.

Bild des Cassiopei Gerätes:

Easy Flash für C64

Von Thomas Giesel (skoe). Homepage: http://skoe.de/easyflash

EasyFlash ist ein Modul für den C64-Expansionsport. Dieses Modul lässt sich im Gegensatz zu klassischen Spielemodulen auch vom C64 aus beschreiben.

Damit kann man leicht diverse klassische Spielemodule, Programmsammlungen mit Menü oder sogar ein Diagnosemodul zur Fehlersuche herstellen. Alles, was man dazu braucht, ist ein C64, ein EasyFlash, die hier erhältliche Software und ein Abbild des Modules (*.crt). Da solche CRT-Dateien recht groß sein können, ist eventuell ein großes Laufwerk wie eine FD-2000 oder ein sd2iec zum Transportieren der CRT-Datei ratsam.

EasyFlash ist kein Freezer-Cartridge wie z.B. das Final Cartridge III oder das Retro Replay. Es ist auch kein Ersatz für ein 1541-Diskettenlaufwerk wie z.B. das sd2iec.

EasyFlash ist ein 1 MByte großes Flash-EPROM-Modul, das verschiedene Speicherkonfigurationen und Bänke unterstützt. Es benutzt keine traditionellen UV-löschbaren EPROMs, sondern Flash-Speicher. Dadurch kann man Module direkt am C64 "brennen". Man braucht dafür kein EPROM-Programmiergerät.

Die EasyFlash-Hardware ist sehr einfach aufzubauen und wird durch ein umfassendes Software-Paket für Benutzer und Entwickler unterstützt. Damit ist es ganz einfach, sich sein eigenes Modul mit den Lieblingsprogrammen darauf zusammenzustellen. Für Entwickler ist es einfach, neue Software für das EasyFlash zu entwickeln oder vorhandene anzupassen.

EasyFlash unterstützt folgende Modulformate: Normal 8k, Normal 16k, Ultimax, Ocean Type 1, EasyFlash.

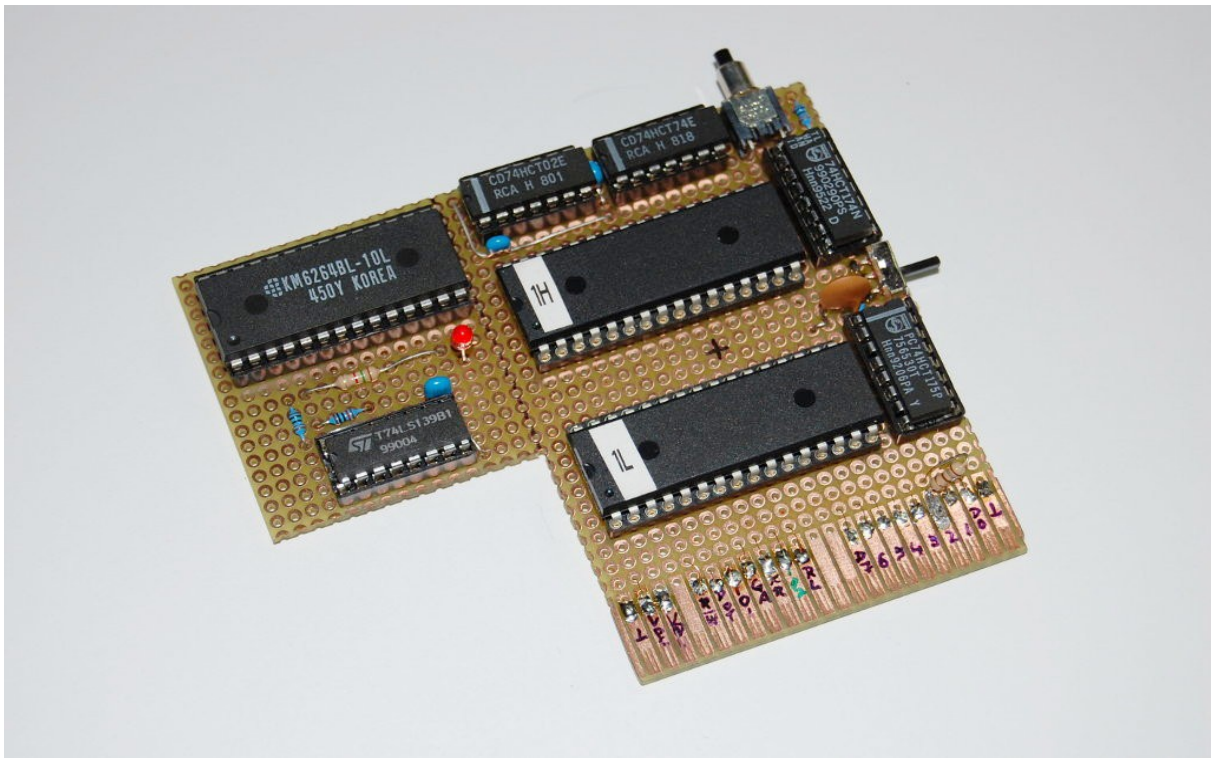
Das Hardware-Design und die Dokumentation ist unter der Creative Commons Lizenz (by-sa) veröffentlicht. Die Software steht unter der zlib-Lizenz. Das heißt, das man damit fast alles machen kann.

Das sind die Teilprojekte von EasyFlash:

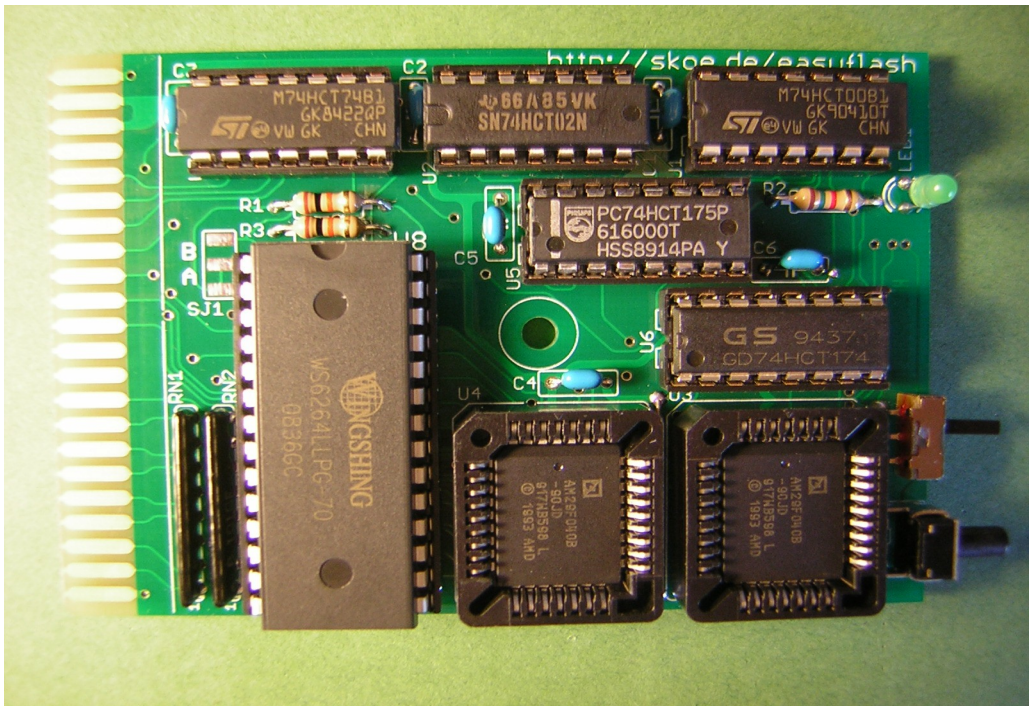
- EasyFlash ist der Name der Hardware
- EasyProg ist ein C64-Programm zum Schreiben von Cartridge-Images (*.crt) auf das Modul
- EasyCRT ist ein PC-Program, mit dem Du Deine eigenen Module-Images erstellen kannst
- EasyLoader ist ein Menü, das zu Deinen Modulen hinzugefügt werden kann
- EasySDK ist eine Sammlung von Dokumenten und Code-Schnipseln für Entwickler

Der erste Prototyp

Hier sieht man den ersten vollständig funktionierenden Prototypen. Er wurde auf einer Lochrasterplatine Rademacher Nr. 944 (Conrad 521250) aufgebaut. Ein Verdrahtungsstift oder ganz normaler Kupferlackdraht sind eine sehr einfache Art, ein EasyFlash-Modul zu bauen. Aber man braucht etwas Geduld :-) Eine perfekt passende Leiterplatte, an der man nicht rumsägen braucht, ist das 8 Bit Baby.



Zur Zeit sind die ersten Platinen für einen Betatest bestellt:



Funktion

Die Adressleitungen A0..A12 für die Flash-Adressen können direkt vom Expansionsport verwendet werden, also von der CPU bzw. vom VIC-II. Damit können pro Flash 8 KiB adressiert werden. Die fehlenden Adressen BA13..BA18 kommen vom Register $DE00, das durch U6 realisiert wird. Damit sind 64 Bänke * 8 KiB * 2 Flashs = 1 MiB möglich.

Nach dem Reset wird durch U5 die Leitung /GAME low und /EXROM high, der C64 aktiviert den Ultimax-Modus. Das bereits erwähnte Banking-Register U6 steht auf 0, so dass Bank 0 des ROMH-Flashs bei $E000..$FFFF sichtbar wird. Ein Reset-Vector in diesem Speicherbereich kann für die weitere Initialisierung sorgen. Durch Schließen von JP1 kann das Aktivieren von /GAME nach dem Reset verhindert werden. In diesem Fall startet der C64 mit unsichtbarem Flash-ROM. Dafür sorgt die Logik in U2A und U2B.

Die Speicherkonfiguration lässt sich über das Register $DE02 (U5) mit beliebigen Kombinationen von /GAME und /EXROM konfigurieren. Sehr praktisch ist, dass trotz gesetztem Jumper die Programmiersoftware die Leitungen /GAME und /EXROM beeinflussen kann. Dadurch kann man die Flashs schreiben, ohne im Betrieb die Stellung des Jumpers bzw. Schalters ändern zu müssen.

/OE für die Flashs wird dann aktiv, wenn Phi2 low ist oder R/W high, das wird durch U1A und U2D erreicht. Diese Verknüpfung wird benötigt, weil auf dem C128 R/W auch low sein kann, wenn der VIC im Ultimax-Modus vom Modulspeicher lesen möchte. Das liegt daran, dass die CPU R/W schon im VIC-Zyklus setzt und dieses Phänomen dort nicht wie beim C64 durch AEC versteckt wird. Die Ausgänge der Flashs werden nur dann tatsächlich durchgeschaltet, wenn zusätzlich zum gerade beschriebenen Signal /OE auch der Eingang /CE für einen Flash-Baustein low ist. Diese Eingänge sind deshalb mit den Leitungen /ROMH bzw. /ROML verbunden, die vom PLA im C64 für die richtigen Adressen auskodiert werden.

Die Erzeugung von /WE ist etwas komplizierter. Direkt nach der H-Flanke von Phi2 sind die Daten-, Adress- und Steuerleitungen von der CPU noch nicht stabil. Das liegt daran, dass der VIC-II erst mit dieser Flanke den Bus wieder freigibt. Deswegen müssen wir uns /WE zu einem späteren Zeitpunkt in diesem Zyklus selbsts generieren. Die beiden Flipflops in U7 sind zu diesem Zweck zu einem 2-Bit Grey-Code-Zähler zusammengeschaltet. Das erste Flipflop (U7B) wird durch Phi2 low im Reset gehalten, das zweite (U7A) durch /OE. Wenn Phi2 high wird, sorgt die erste darauf folgende H-Flanke von Dotclock dafür, dass Q von U7B high wird. Damit diese erste Flanke von Dotclock garantiert nicht verpasst wird, benutzen wir für U7B direkt Phi2 als Reset-Signal. Warum das beim zweiten anders ist, wird gleich klarer werden.

Bei der zweiten H-Flanke kann Q von U7B auf Q von U2A durchgeschaltet werden. Die geschieht aber nur, wenn /OE zu diesem Zeitpunkt high ist, da dieses sonst U7A im Reset halten würde. Wir sehen also, das U7 nicht nur für das Generieren des Write-Timing benutzt wird, sondern auch für

das Verhindern eines /WE-Signals bei Lesezugriffen. Falls es sich nicht um einen Lesezugriff handelt, wird jetzt /Q von U7A low, also /WE aktiv.

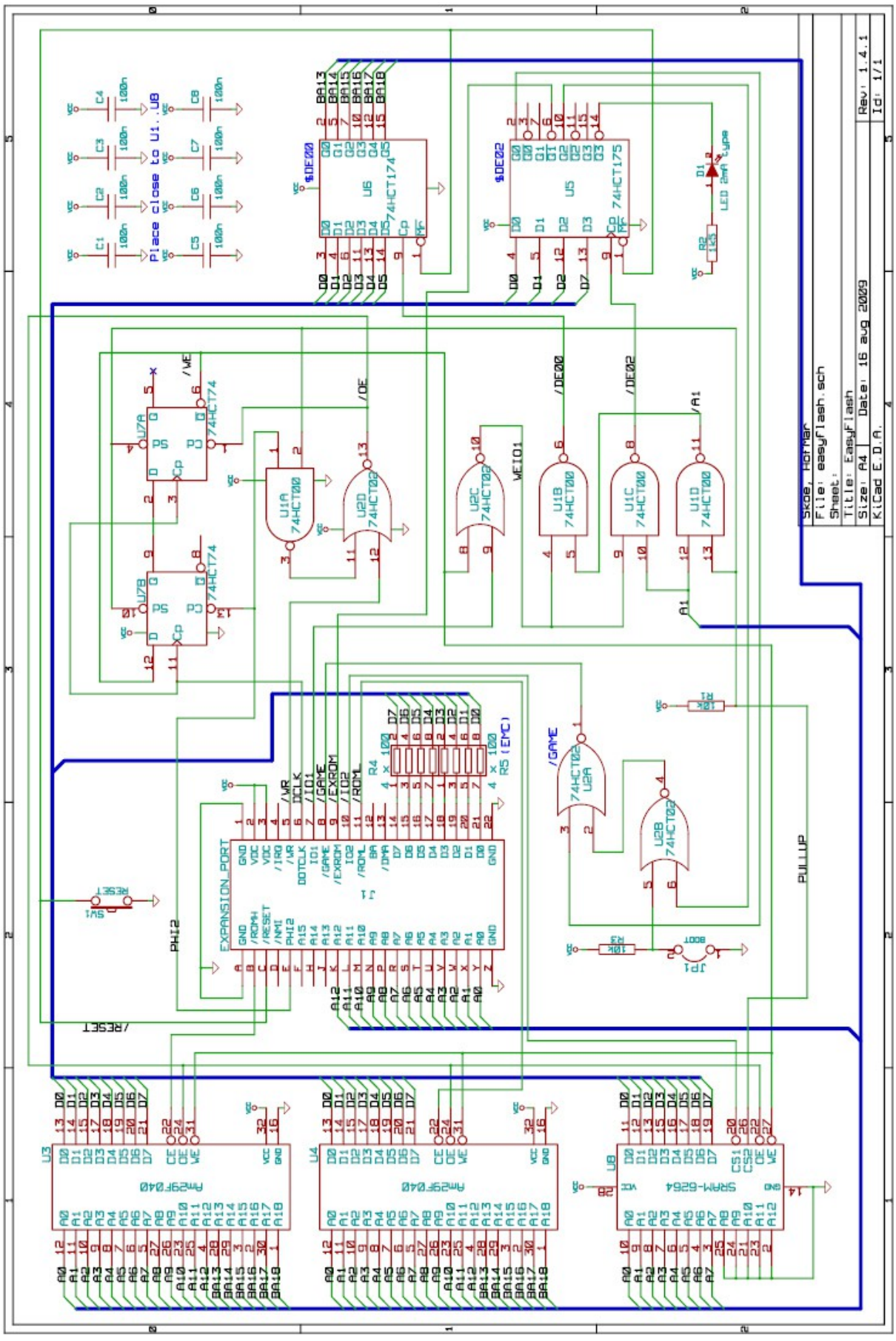
Bei der dritten H-Flanke von Dotclock ändert sich nichts am Zustand der /WE-Leitung. Erst bei der vierten wird durch die Beschaltung als Grey-Code-Zähler /WE wieder high, rechtzeitig kurz vor dem Ende des Phi2-Zyklus. Das so gewonnene Schreibsignal wird sowohl für die Flashs benutzt, als auch für die beiden Register und den RAM.

U2C und U1B/C/D sorgen durch die Verknüpfung von /WE, /IO1 und der Adressleitung A1 dafür, dass die beiden Register bei Schreibzugriffen auf die richtigen Adressen beschrieben werden.

Vom RAM-Baustein U8 sind 256 Bytes an der C64-Adresse $DF00 zu sehen. Dafür kann das Signal /IO2 direkt als /CS benutzt werden. Die bereits beschriebenen Signale /OE und /WE sorgen für die korrekte Steuerung dieses Bausteins. Aus Sicht der Hardware wäre es naheliegend, die zusätzlichen Adressleitungen aus dem Bank-Register U6 für Banking des RAMs zu benutzen. Von dieser Möglichkeit wurde jedoch aus Gründen der Software-Entwicklung abgesehen: Wenn dieser Speicher für eine kleine bankübergreifende Leseroutine benutzt wird, darf er keinesfalls beim Umschalten der Bank ausgetauscht werden.

Es sei noch erwähnt, dass die Leitungen /ROML und /ROMH im normalen C64-Modus nur bei Lesezugriffen aktiv werden. Im Ultimax-Modus hingegen werden sie auch für Schreibzugriffe auf die entsprechenden Adressen auskodiert. Deswegen lassen sich die Flash-Bausteine nur im Ultimax-Modus beschreiben. Programmiersoftware wie z.B. das Programm EasyProg muss diesen Umstand berücksichtigen.

Schaltplan:


Place close to U1..U8
EXPANSION_PORT
PULLUP
Skoe, HdMar
File: easyflash.sch
Sheet:
Title: EasyFlash
Size: A4
Date: 16 aug 2009
Rev: 1.4.1
Id: 1/1
KiCad E.D.A.

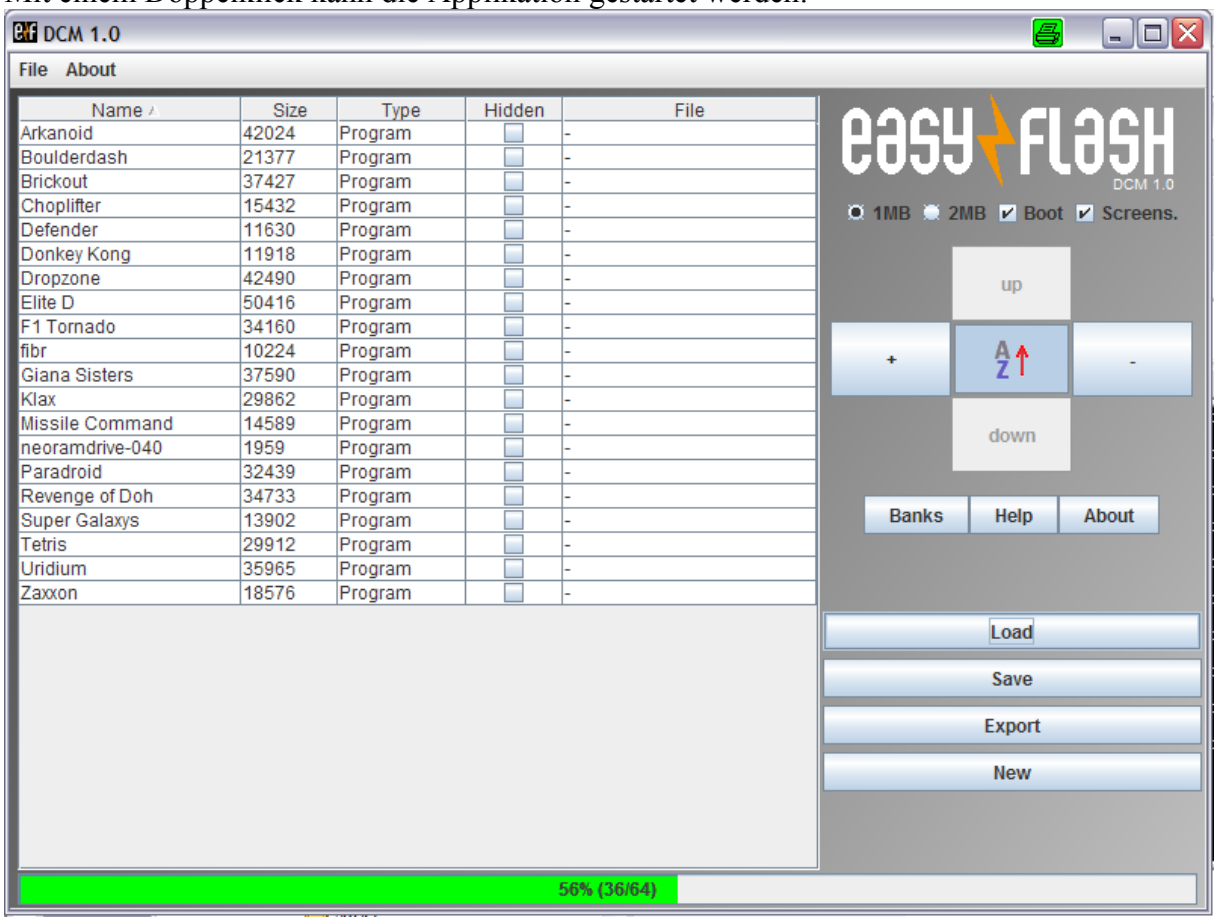
DCM – Draco Cart Maker

Link: http://www.sascha-bader.de/html/dcm

Von Sascha Bader wurde ein Programm entwickelt um sich einfach selbst eine Cartridgedatei für das Easyflash zusammen zu stellen. Das Programm ist plattformübergreifend in Java geschrieben. Unter Windows dazu am besten eine kleine Batchdatei ‚dcm.bat' erstellen:

java -jar dcm.jar

Mit einem Doppelklick kann die Applikation gestartet werden:



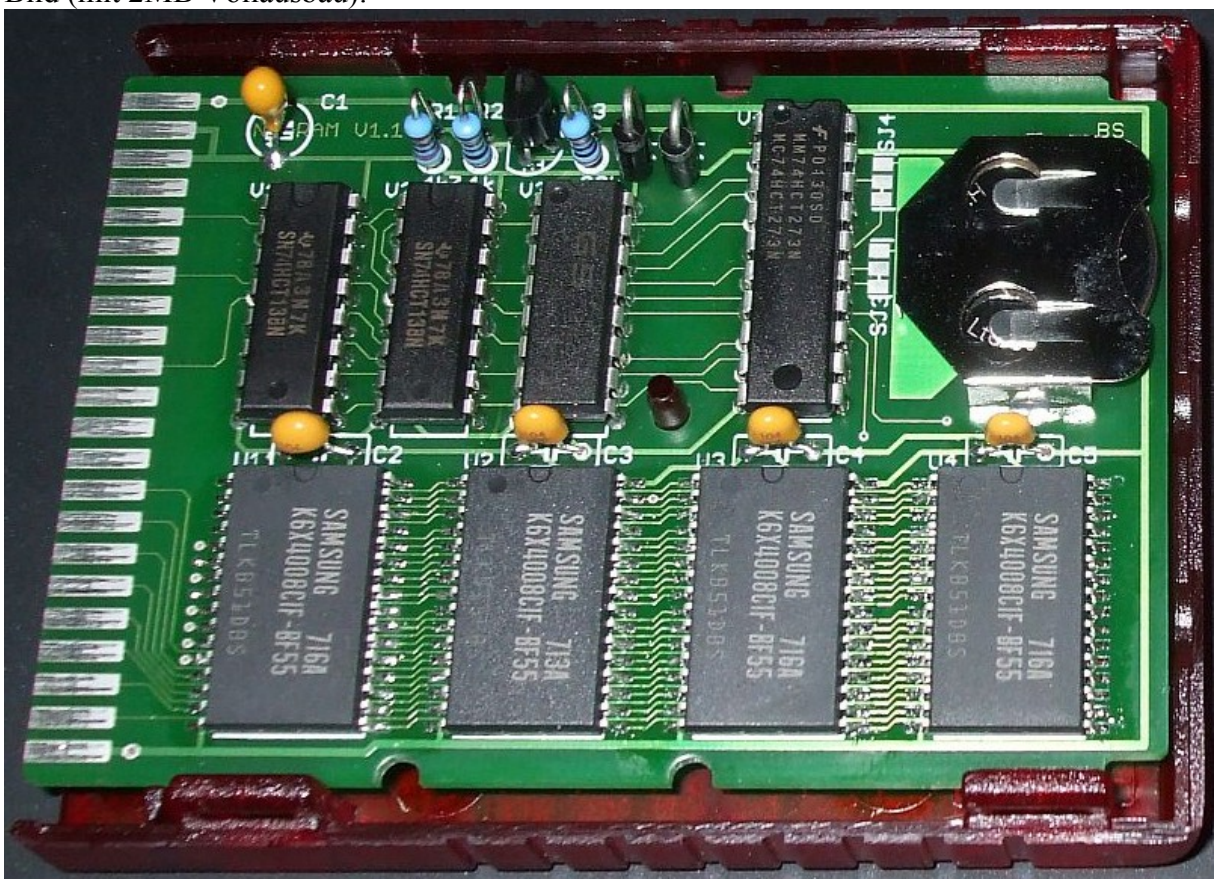
Über ‚Load' kann eine vorhande Cartridgedatei geladen werden.
‚Save' sichert die Zusammenstellung in einer Cartridgedatei.
Über ‚+' können Programme zugefügt werden; über ‚-' werden sie entfernt.
Bei nicht alphabethischer Sortierung, können einzelne Programme über ‚up' und ‚down' in den Programmplätzen verschoben werden (nach open bzw. unten bewegt werden).

NeoRAM - Batterie gepufferte RAM Disk für C64

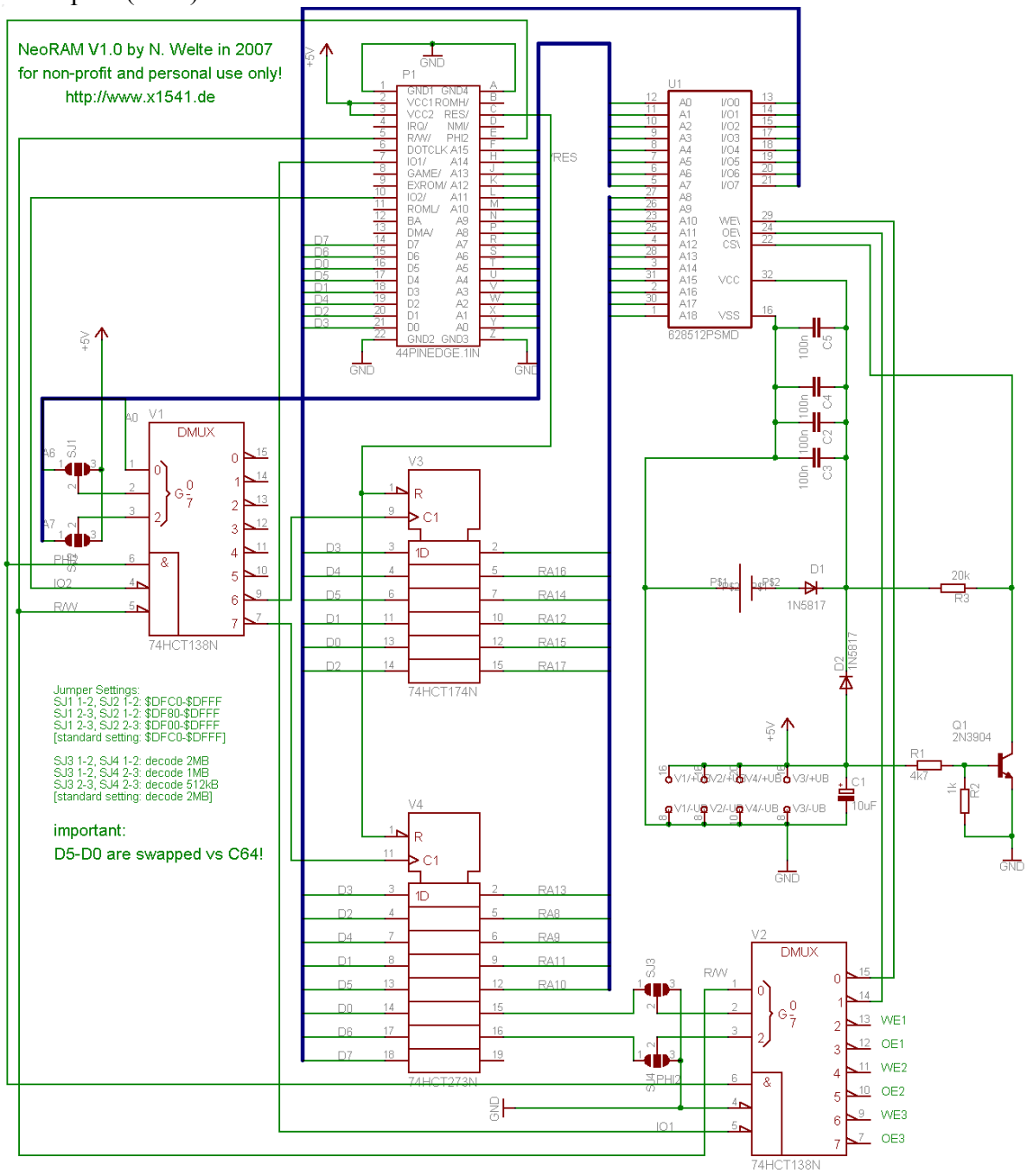
Diese Akku-gepufferte Ram Karte wurde entwickelt von Nicolas Welte (x1541). Sie ist als Klassiker zu bezeichnen, weil sie ohne AVR, CPLD, GAL etc. auskommt und nur Standardtechnik und Bausteine verwendet. Die NeoRAM ist zur GeoRAM kompatibel. Der RAM kann in 512k Schritten ausgebaut werden von 512k bis zu 2 Megabyte. Durch die Pufferung mit einer Lithium-Zelle bleiben die Daten für ca. 2-3 Jahre erhalten. Wenn man die Batterie wechselt, während die Ram Karte vom C64 mit Strom versorgt wird, sollten die Daten auch den Wechsel unbeschadet überstehen.

Mehr Informationen gibt es im Forum64 oder auf Nicolas Homepage: http://www.x1541.de

Bild (mit 2MB Vollausbau):

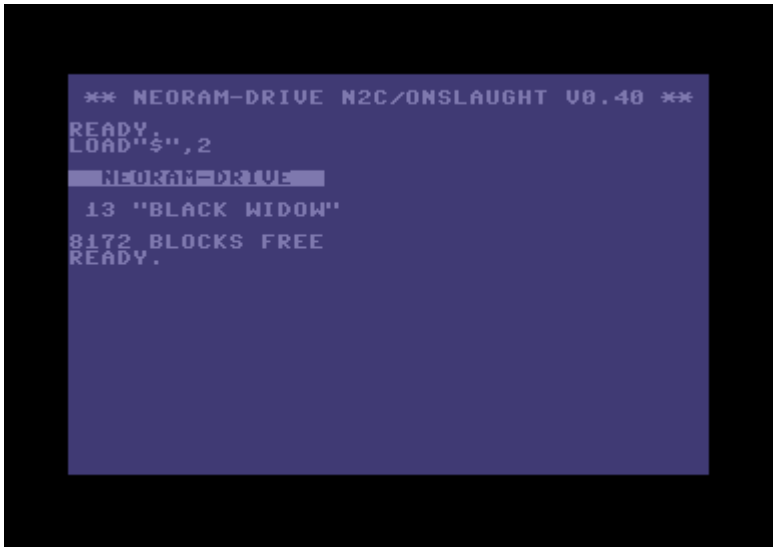


Schaltplan (512k):



Für mich wurde die Karte erst besonders interessant, durch den Ramdisk-Treiber von enthusi (Martin Wendt) aus dem Forum64. Damit kann die Karte wie ein Laufwerk angesprochen werden. Die Ladezeiten (siehe Speedmessungen Kapitel) sind mit Faktor 20 ebenfalls erfreulich hoch. Das tolle daran ist, das man mit einer initialisierten Ram Karte zu einem anderen C64 gehen kann und ohne Zusatzsoftware, einfach durch SYS 57000 die Ramdisk einbinden kann! Schon kann man mit load"$",2:list die gespeicherten Pogramme sehen und natürlich auch laden. Super!

Den Ramdisk Treiber gibts in der CSDB: http://noname.c64.org/csdb/release/index.php?id=61122



Speedmessungen verschiedener Lösungen

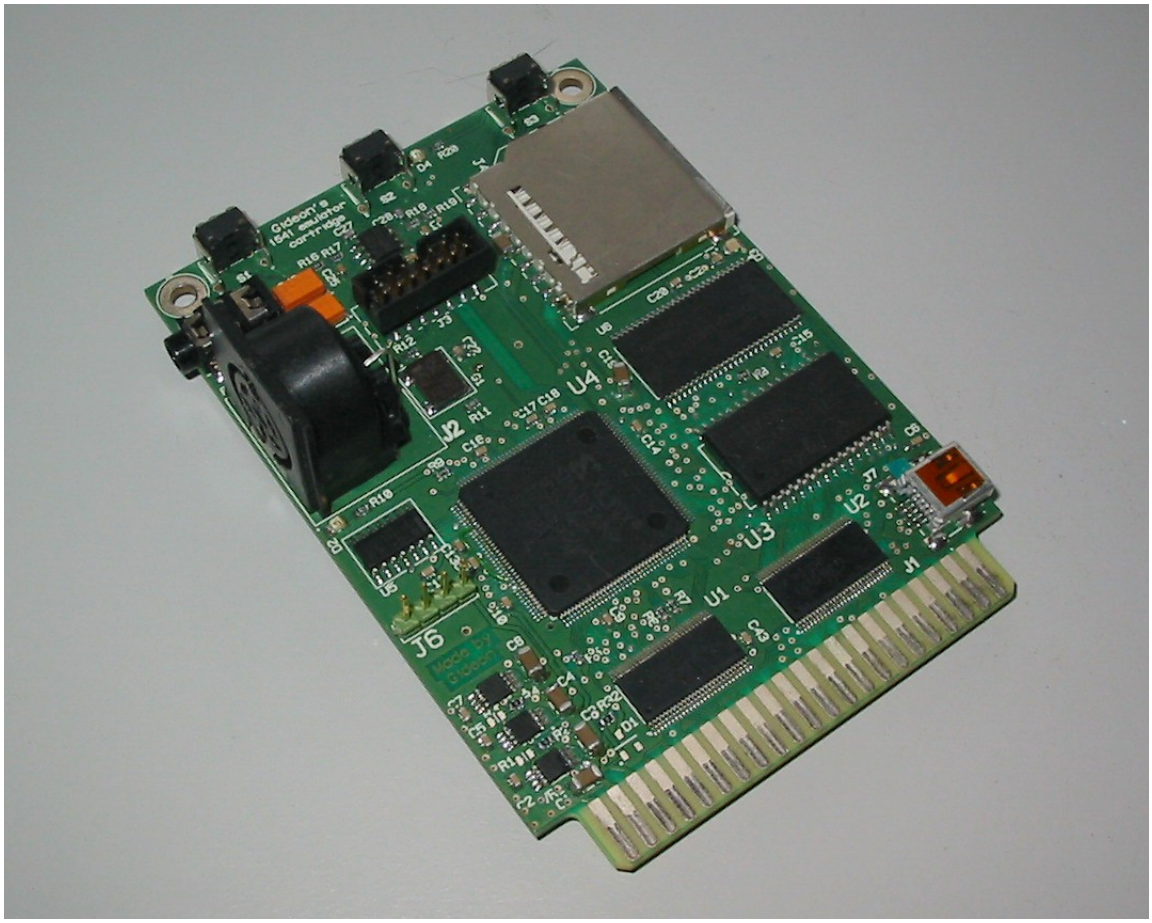
Die hier gezeigten Messungen wurde von unterschiedlichen Usern des Forum64 durchgeführt und sind hier nach den Programm Ladezeiten sortiert:

Testumgebung	Progr. SAVE	Progr. LOAD	SEQ schreiben	SEQ lesen
C64+IDE64 V3.4 Firmware mit SCPU Unterstützung.	456,67	635,00	61,43	76,00
Ramlink mit Jiffy	47,24	115,45	14,58	13,33
Ramlink mit Fastload	47,24	105,83	16,86	16,17
CMD HD/1541 Partition	37,03	52,92	17,20	19,49
Dolphin 2+Dolphin 2 1541	14,89	25,92	6,10	9,05
Vice 1.22+1541 Emulation	18,77	24,42	34,40	44,71
SX 64 + Professional DOS+1541	13,05	23,09	4,73	4,29
Jiffy+sd2iec~0.5+1541-II	**7,29**	**21,53**	**10,00**	**12,26**
Jiffy+1581	6,37	19,54	8,11	11,18
C64+NeoRAM Ramdrive 0.3	**18,77**	**18,68**	**2,54**	**1,63**
SpeedDos+AR6+SpeedDos 1541	8,51	18,41	1,67	3,49
Action Replay 4.2 Professional	8,56	18,14	1,22	1,16
Jiffy+AR6+Jiffy 1541	8,84	18,14	2,78	2,74
Dolphin 2+AR6+Dolphin 2 1541	7,21	18,14	6,52	11,18
64'erV4+AR6+64'erV3 1541	7,21	18,14	1,70	3,25
AR6+CBM 1541	8,51	17,89	1,22	1,16
128D (64er Mode) interne 1571 AR5 '88	8,62	17,89	1,22	1,16
AT7.5+Jiffy+1581	11,30	16,70	7,89	11,34
Magic Formel 2.0 + 1541	14,89	16,28	5,70	5,51
AR7.5+1581	11,00	16,00	2,40	1,34
AR7.5+1571	7,00	15,50	1,00	1,00
AR7.5+Jiffy+1571	7,65	15,49	2,30	6,91
FSD+AR6+Jiffy 1581	8,95	15,49	4,94	6,03
Jiffy+AR6+Jiffy 1581	8,95	15,49	7,29	11,34
AR7.5+1541	7,00	15,30	1,00	1,00
AR6+Jiffy 1581	8,78	15,12	2,61	1,54
FD2000 mit Jiffy	5,32	15,12	6,37	8,94
Jiffy+Jiffy 1581	5,07	14,77	6,19	9,38
Jiffy+1571	2,77	14,43	2,35	6,18
FSD+Jiffy 1581	3,69	14,43	4,22	5,03
Basic Jiffydos Fastload+FD-2000	20,45	13,96	6,24	9,16
Action Power Cartr. 8.1DE	6,49	13,66	0,94	0,99
Turboload+sd2iec 0.5.0 - LarsP Aufbau	2,26	12,83	2,56	1,63
Beast System+CBM 1541	1,00	11,04	0,99	1,00
EXOS V3+CBM 1541	1,00	11,04	0,99	1,00
128D (64er Mode) int.1571 mit RR	0,98	10,50	0,99	0,96
FC3+Jiffy+1541	2,60	10,40	2,20	5,31
FC3+Jiffy+1571	2,80	10,24	2,35	6,23
Jiffy+1541	2,54	9,84	2,20	5,30
C64+OC118 (beide mii Jiffydos)	2,50	9,77	2,13	5,17
Jiffy+Jiffy 1541	2,49	9,62	2,12	5,21
FC3+1541	5,40	9,34	1,00	1,00
FC3+1571	5,33	9,34	1,00	1,00
FC3+1541	5,76	9,07	1,00	1,00
FC3+CBM 1541	5,71	9,07	1,00	1,00
SpeedDos+SpeedDos 1541	1,34	8,82	1,39	3,04

1541 Ultimate

Das MMC2IEC verhält sich gegenüber einem C64 wie eine 1541. Dabei wird die IEC Schnittstelle abgebildet. Ansonsten hat MMC2IEC halt nicht viel gemeinsam mit einer 1541 Floppy. Diese hatte eine eigene 6502 CPU, Ram und Rom – war also ein eigener kleiner Computer.

Die 1541 Ultimate von Gideon Zweijtzer emuliert eine 1541 fast exakt mittels Emulation der einzelnen Chips in FPGA.



Die 1541U ist ein Modul für die Commodore 8-Bit Computer. Es ist sowohl eine Floppy 1541 als auch ein Modul wie Action Replay oder Final Cartrige III. Das Herzstück der Platine ist ein mit 50MHz getakteter FPGA. Zusätzlich befinden sich ein SD-Karten Steckplatz, 2MB Flashrom und 512kB SRAM auf der Platine. In der Plus Version gibt es auch noch 32MB SDRAM, die unter anderem als RAM-Laufwerk kompatibel zur REU genutzt werden können. Das besondere der 1541 Ultimate ist, dass sie nicht nur das IEC-Protokoll der seriellen Schnittstelle von Commodore nachbildet, sondern eine echte 1541 emuliert. So können selbst Nachlader-Programme mit eigenen Schnellladeroutinen ohne Probleme von SD-Karte gestartet werden.

Die 1541 Ultimate besitzt zwei Betriebsmodi:

Der Normalbetrieb:

Hier wird die 1541U einfach in den Expansionsport des C64 oder C128 gesteckt und mit dem Seriellen-Kabel an der IEC-Buchse des Computers verbunden. Natürlich können gleichzeitig auch noch andere Laufwerke angeschlossen werden. Die 1541U hat drei Tasten. Es gibt eine Reset- und eine Freezertaste die entsprechend dem emulierten Modul funktioniert. Zusätzlich gibt es eine dritte Taste, mit der man in den Datei-Browser gelangt. Hier kann man durch die Verzeichnisse der SD-Karte wechseln, Disketten-Images im D64-Format mounten, neue Verzeichnisse oder D64-Dateien erstellen und PRG-Dateien oder Programme aus einem D64-Image heraus per DMA in den Speicher laden und ausführen.
Bei der Benutzung mit einem C128 ist dabei zu beachten, dass der Datei-Browser im 40-Zeichen Modus läuft. Falls man also gerade im 80-Zeichen Modus arbeitet, muss man für den Datei-Browser kurzzeitig den Monitor auf 40-Zeichen umstellen.
Die Bedienung ist sehr einfach und der Datei-Browser steht jederzeit zur Verfügung. Wenn ein Programm zum Beispiel einen Diskettenwechsel verlangt, reicht ein kurzer Druck auf die Taste, dann wählt man die entsprechende D64-Datei aus und weiter geht es.

Der Stand-Alone-Betrieb:

Hier wird die 1541U nur über das serielle Kabel mit einem Commodore Rechner verbunden. Für die Stromversorgung benötigt man noch ein Netzteil mit einem Mini-USB-Anschluß. Man kann die 1541U auch einfach mit einem USB-Kabel mit einem PC verbinden. Diese Verbindung dient nur zur Stromversorgung. Ein Datenaustausch über die USB-Schnittstelle ist nicht möglich. In diesem Modus kann man die 1541U „blind“ über die drei Taster steuern. Man kann mit den drei Tasten zur nächsten oder zur vorherigen Datei wechseln oder eine Datei auswählen. Es gibt aber noch eine zweite Möglichkeit der Steuerung im Stand-Alone-Modus. Über eine eigene ID (Standart mäßig ID9) kann man mit DOS-Befehlen direkt auf die SD-Karte zugreifen. Wie im Dateibrowser kann man auch hier durch die Verzeichnisse wechseln, D64-Images mounten oder auch Programme direkt laden und starten.
Der Stand-Alone-Betrieb ist zwar nicht so komfortabel, bietet aber den Vorteil, dass man die 1541U nicht nur am C64 und C128 betreiben kann, sondern auch an allen anderen Commodore Rechnern die über einen IEC-Bus verfügen (VC-20, C16, …). Auch wenn man einmal ein anderes Modul an seinem C64 oder C128 benutzen möchte, brauch man so nicht auf die 1541 Ultimate verzichten.

Um die Aktivitäten der 1541 Ultimate sehen zu können, gibt es 3 LEDs. Die eine zeigt den Zugriff auf die SD-Karte an, die zweite zeigt an, ob ein D64-Image geladen ist und die letzte zeigt den Betrieb der emulierten Floppy (wie die LED an einer echten 1541).

Mehr Informationen zu diesem Projekt gibts hier:

http://www.1541ultimate.net

Ersatz für den 8701 IC

Von Thomas Tahsin-Bey

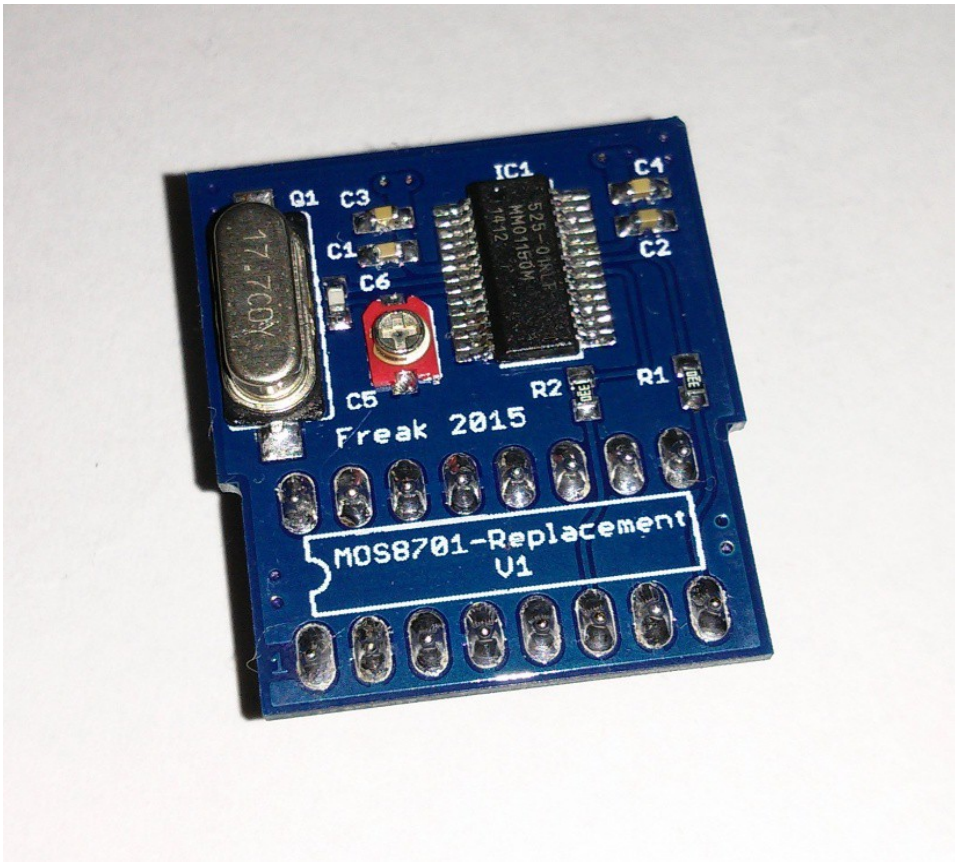
Links:
http://clone-a-dore.blogspot.de/
http://www.forum64.de/wbb3/board65-neue-hardware/board289-diverses/board293-sonstiges-ohne-eigene-rubrik/61322-kleine-ersatzplatine-f-r-den-8701

Der 8701 Chip ist ein Clock Generator IC, der im C64 verbaut wurde. David Maurer hat auf seiner Internetseite "Clone-A-Dore" im August 2014 einen Ersatzchip für den 8701 vorgestellt, der offensichtlich den IC ICS525-01 von IDT verwendet.Ich habe aus den mir zur Verfügung stehenden Informationen eine Ersatzschaltung aufgebaut.

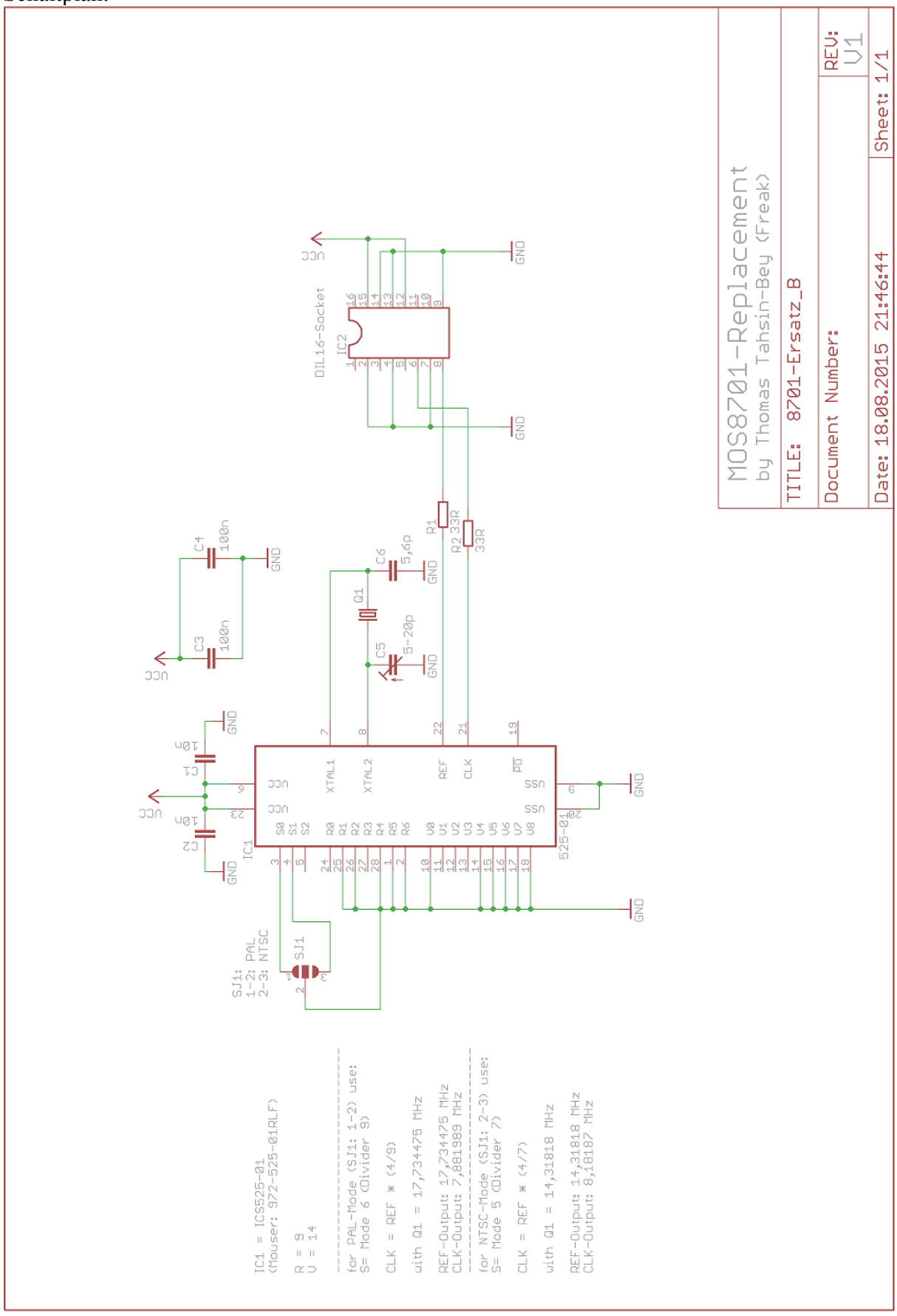
Original Pinout (app.):

N/C	1	16	N/C
Output Enable	2	15	+5V for Crystal
N/C	3	14	Crystal In
Disable pin 11	4	13	Crystal Out
- Reset	5	12	Vcc
Dot Clock	6	11	Unknown Output
Enable PAL	7	10	+5V Constant Output
Color Clock	8	9	Vss

Platine:



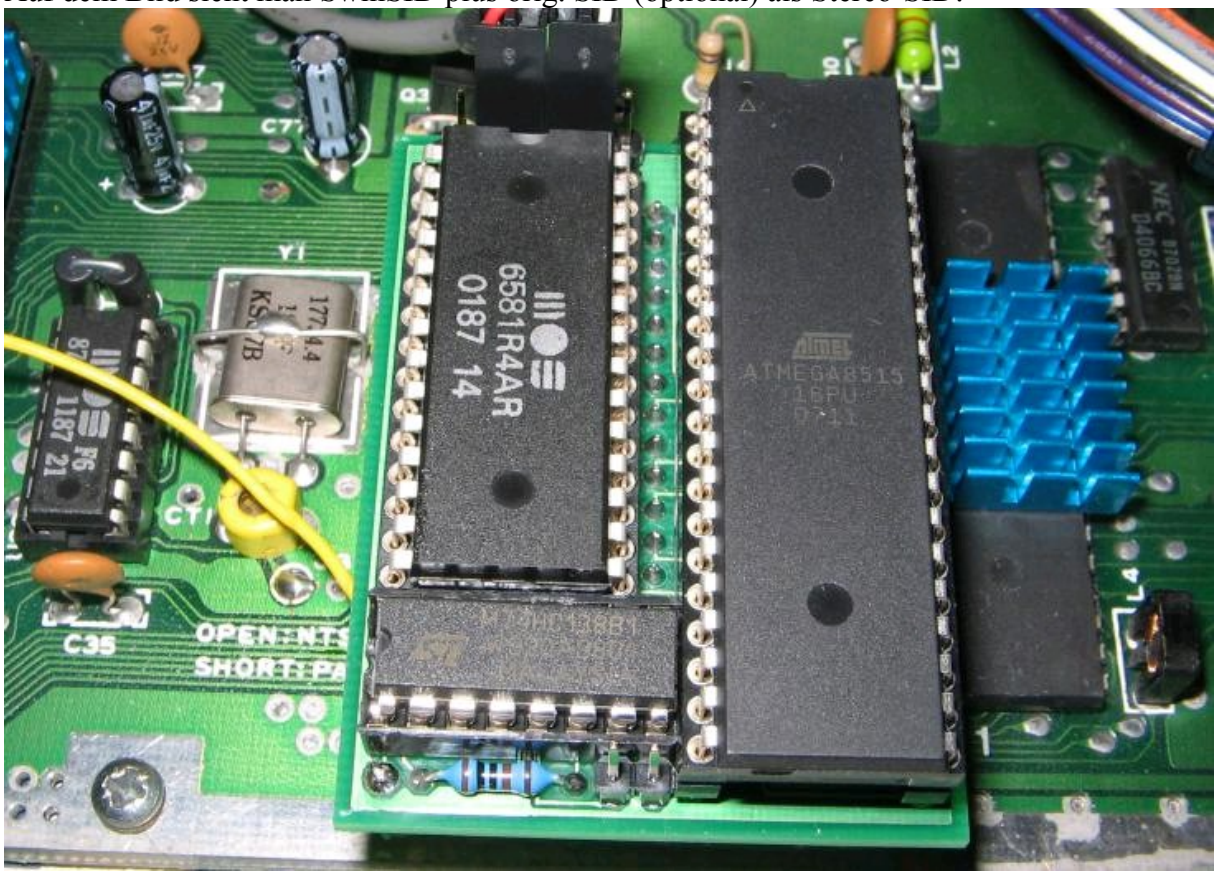
Schaltplan:


MOS8701-Replacement
by Thomas Tahsin-Bey (Freak)
TITLE: 8701-Ersatz_B
Document Number:
REV: V1
Date: 18.08.2015 21:46:44
Sheet: 1/1
IC1 = ICS525-01
(Mouser: 972-525-01RLF)
R = 9
V = 14
for PAL-Mode (SJ1: 1-2) use:
S= Mode 6 (Divider 9)
CLK = REF * (4/9)
with Q1 = 17,734475 MHz
REF-Output: 17,734475 MHz
CLK-Output: 7,881989 MHz
for NTSC-Mode (SJ1: 2-3) use:
S= Mode 5 (Divider 7)
CLK = REF * (4/7)
with Q1 = 14,31818 MHz
REF-Output: 14,31818 MHz
CLK-Output: 8,18187 MHz
SJ1:
1-2: PAL
2-3: NTSC
IC1
525-01
IC2
DIL16-Socket
Q1
C1 10n
C2 10n
C3 100n
C4 100n
C5 5-20p
C6 5,6p
R1 33R
R2 33R
GND
VCC

SwinSID – SID Ersatz auf AVR Basis

Auf Basis AVR Atmega8515 konnte ein Ersatz für den schwer beschaffbaren Soundchip SID 6581 geschaffen werden. Die ersten Exemplare sind inzwischen von Forum64 Mitgliedern nachgebaut worden und die Ergebnisse sehr vielversprechend.

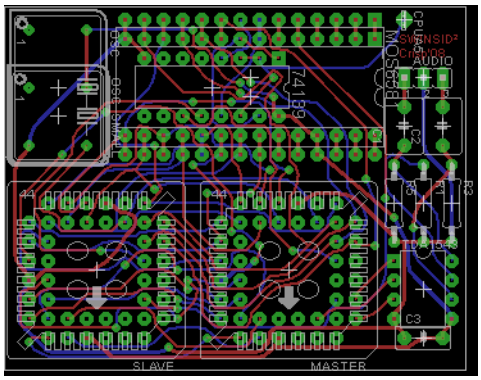
Auf dem Bild sieht man SwinSID plus orig. SID (optional) als Stereo-SID:



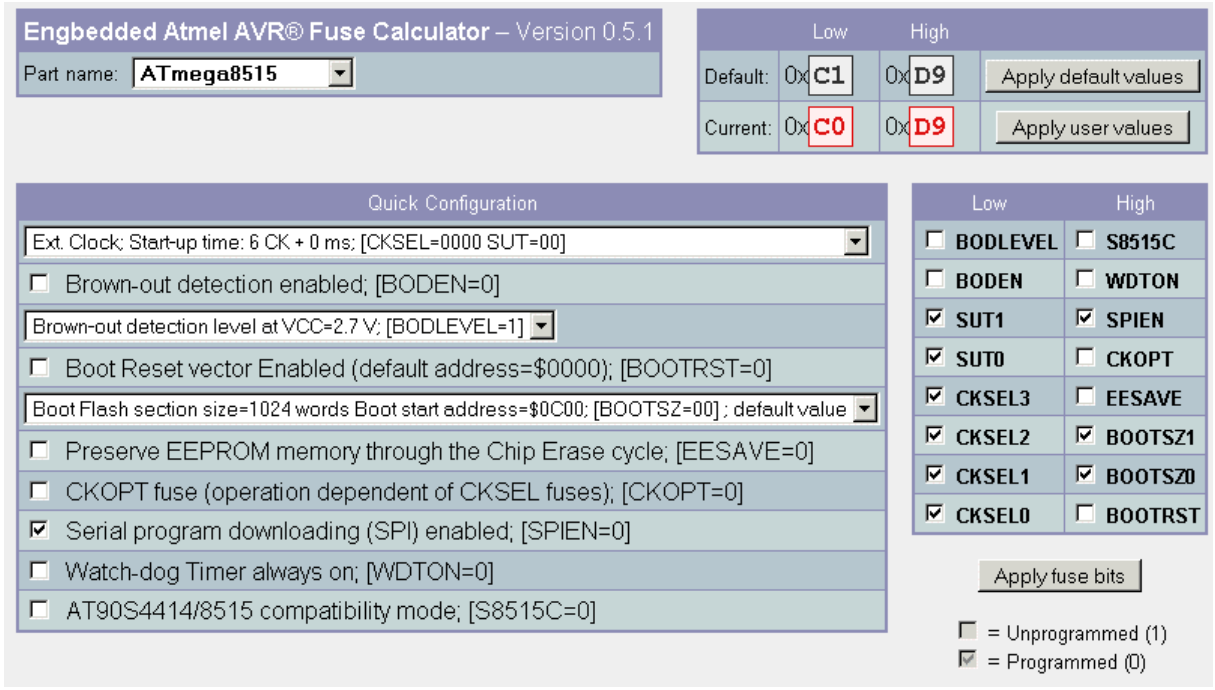
Schaltplan und Firmware für den AVR gibts auf der Homepage:

http://www.swinkels.tvtom.pl/swinsid/

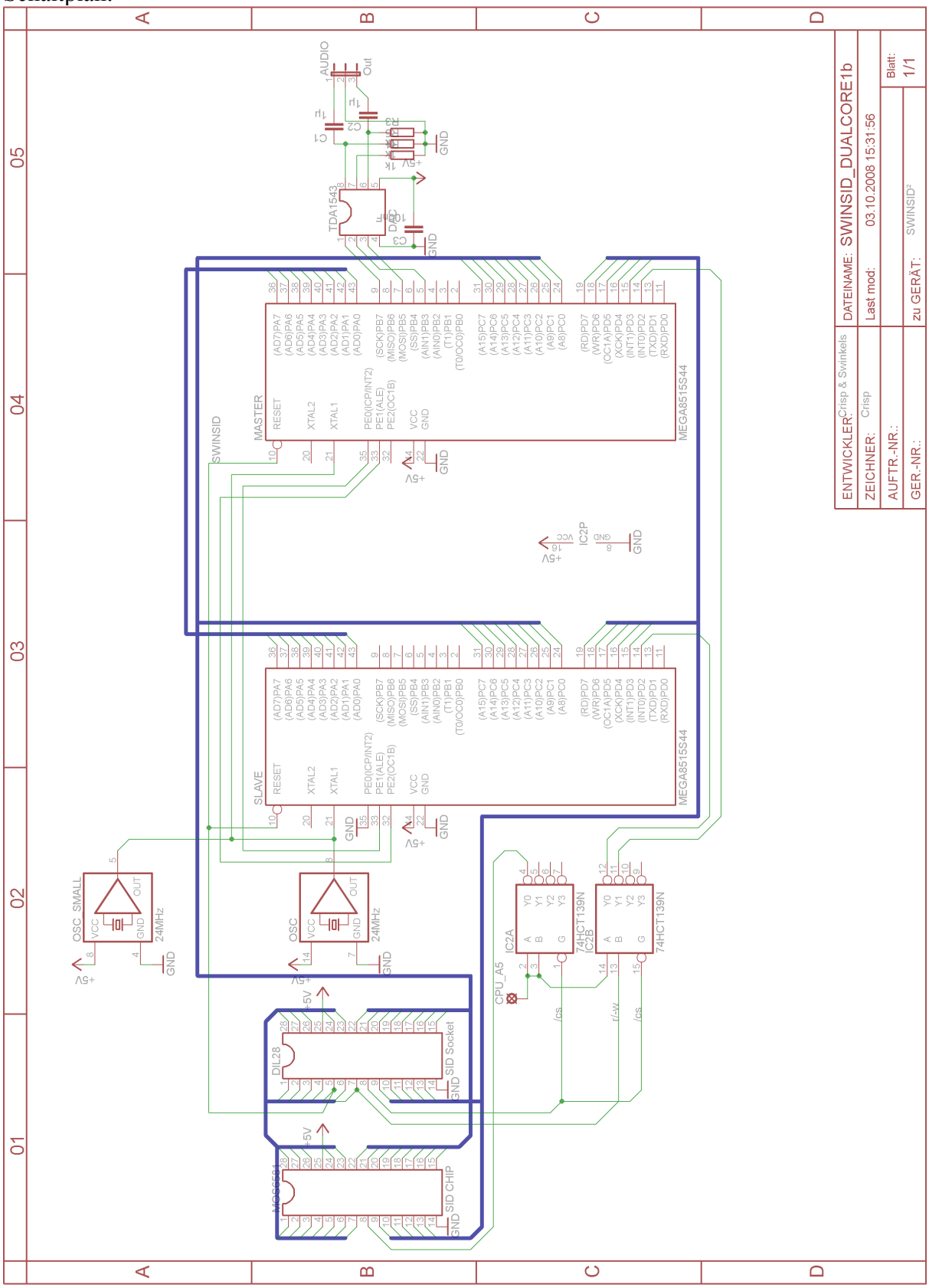
Ingmar Weigel aus dem forum64 hat die Schaltung auf 2 AVR's erweitert und eine Platine dazu geroutet (SwinSIDX2):



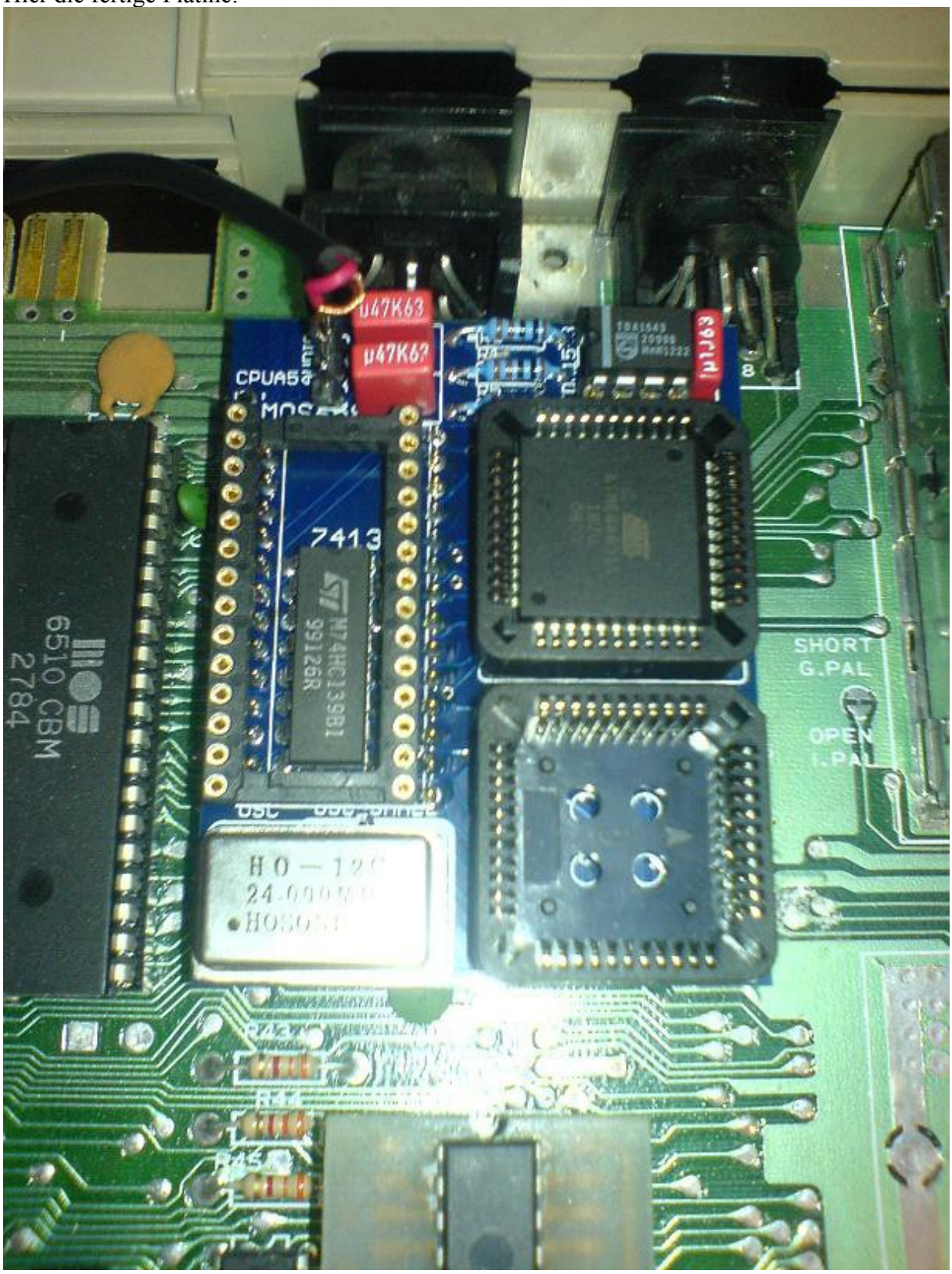
Fuses:



Schaltplan:

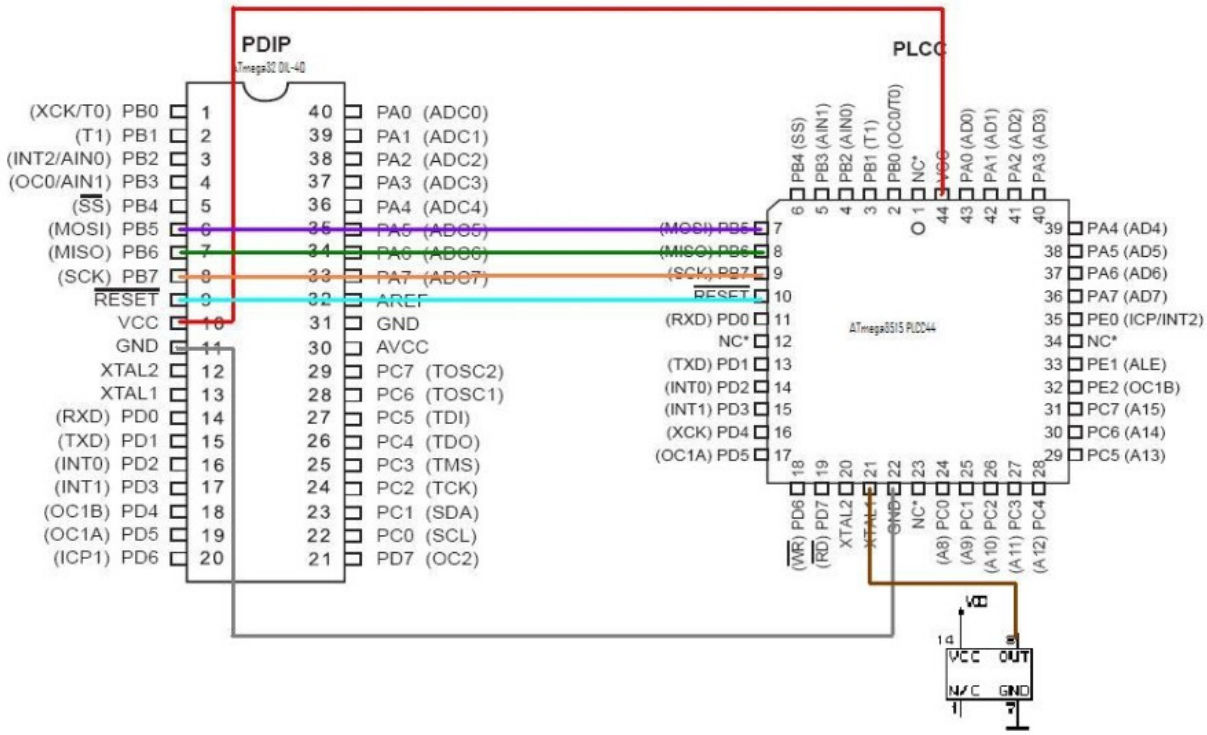

ENTWICKLER: Crisp & Swinkels
ZEICHNER: Crisp
AUFTR.-NR.:
GER.-NR.:
DATEINAME: SWINSID_DUALCORE1b
Last mod: 03.10.2008 15:31:56
zu GERÄT: SWINSID²
Blatt: 1/1
MASTER
SLAVE
SWINSID
MEGA8515S44
TDA1543
OSC SMALL
OSC
24MHz
DIL28
SID Socket
SID CHIP
74HCT139N
CPU_A5
+5V
GND

Hier die fertige Platine:

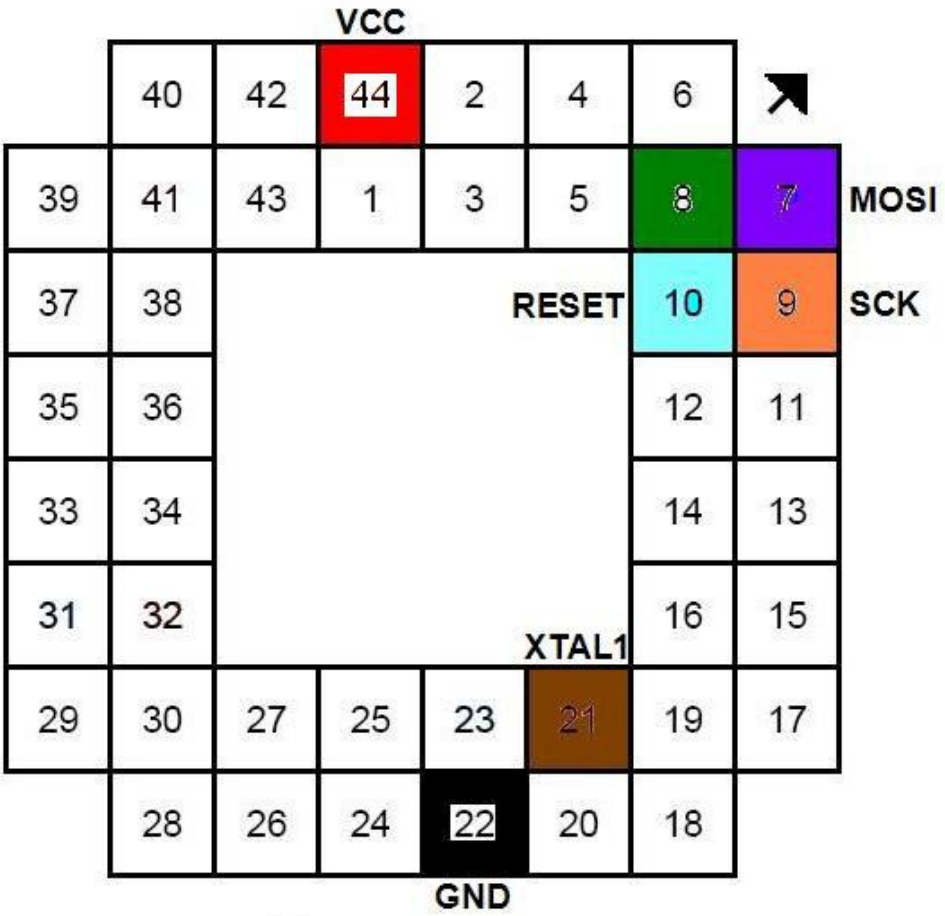


Im Mono-Betrieb als SID Ersatz. Slave-AVR kann dazu unbestückt bleiben und Leitung zur CPU A5 wird ebenfalls nicht benötigt. Sound-Out ist an Pin 3 der Audioleiste.

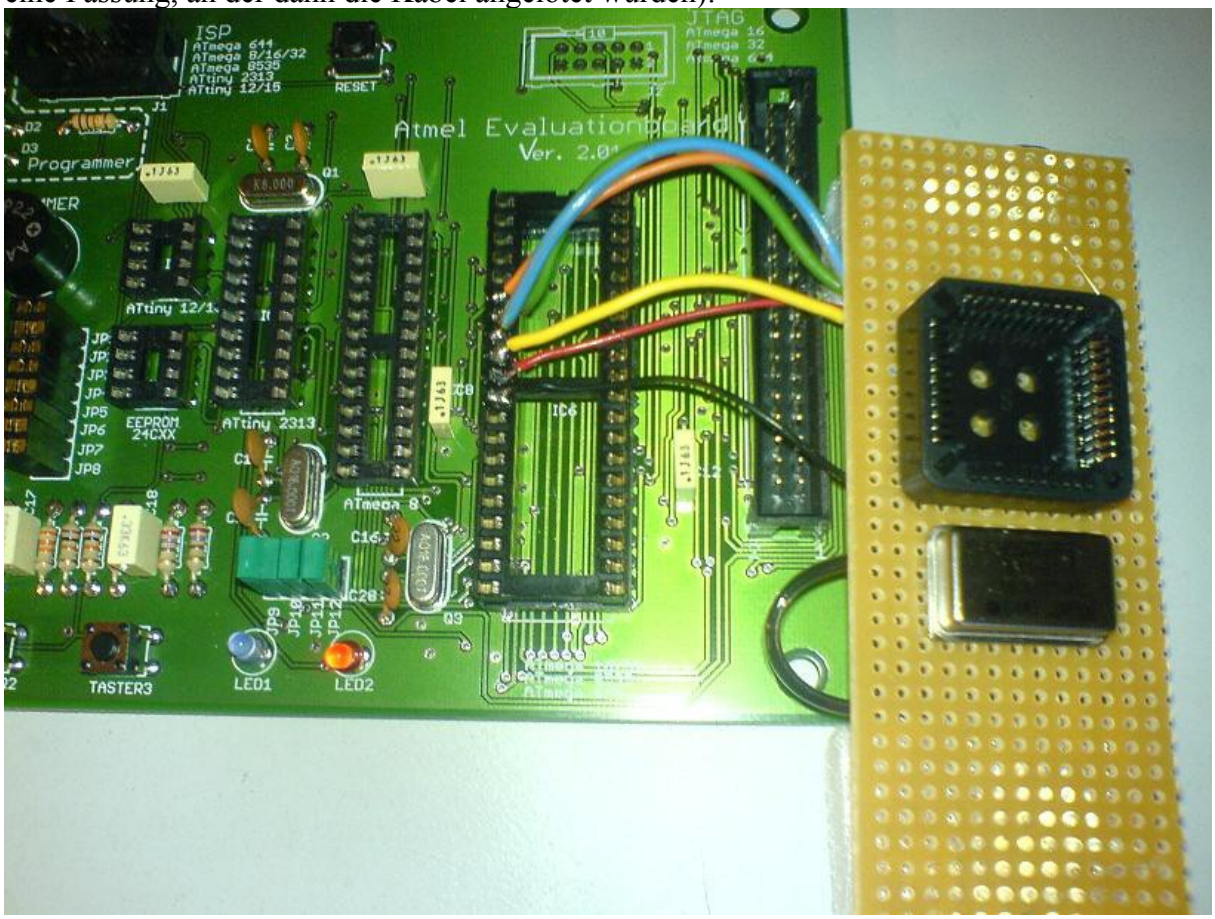
Zum programmieren im Pollin Board kann man folgenden DIL-40 Adapter für die Atmega32 Fassung bauen:



PLCC 44 Fassung von unten



Hier ein Bild eines aufgebauten Adapters (In der Atmega32 Fassung steckt natürlich auch eine Fassung, an der dann die Kabel angelötet wurden):



Micro SwinSID88

Endlich im Oktober 2010 geht ein kleiner Traum in Erfüllung. Ein preisgünstiger SID Ersatz auf AVR Basis, der direkt in den SID Slot gesteckt wird.

Slawomir Musial (Swinkels) und Ingmar Weigel haben den TDA Chip durch Nutzung von Pulsweitenmodulation (PWM) ersetzen können.

Zum Einsatz kommt ein Atmega88PU, der über einen 32MHz Quarzoszillator übertaktet wird.

Fuses: lfuse:0xe0 hfuse:0xdf

Layout:

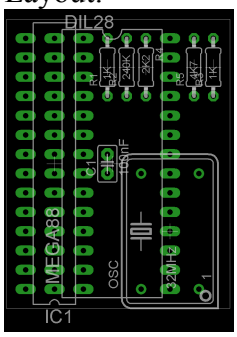
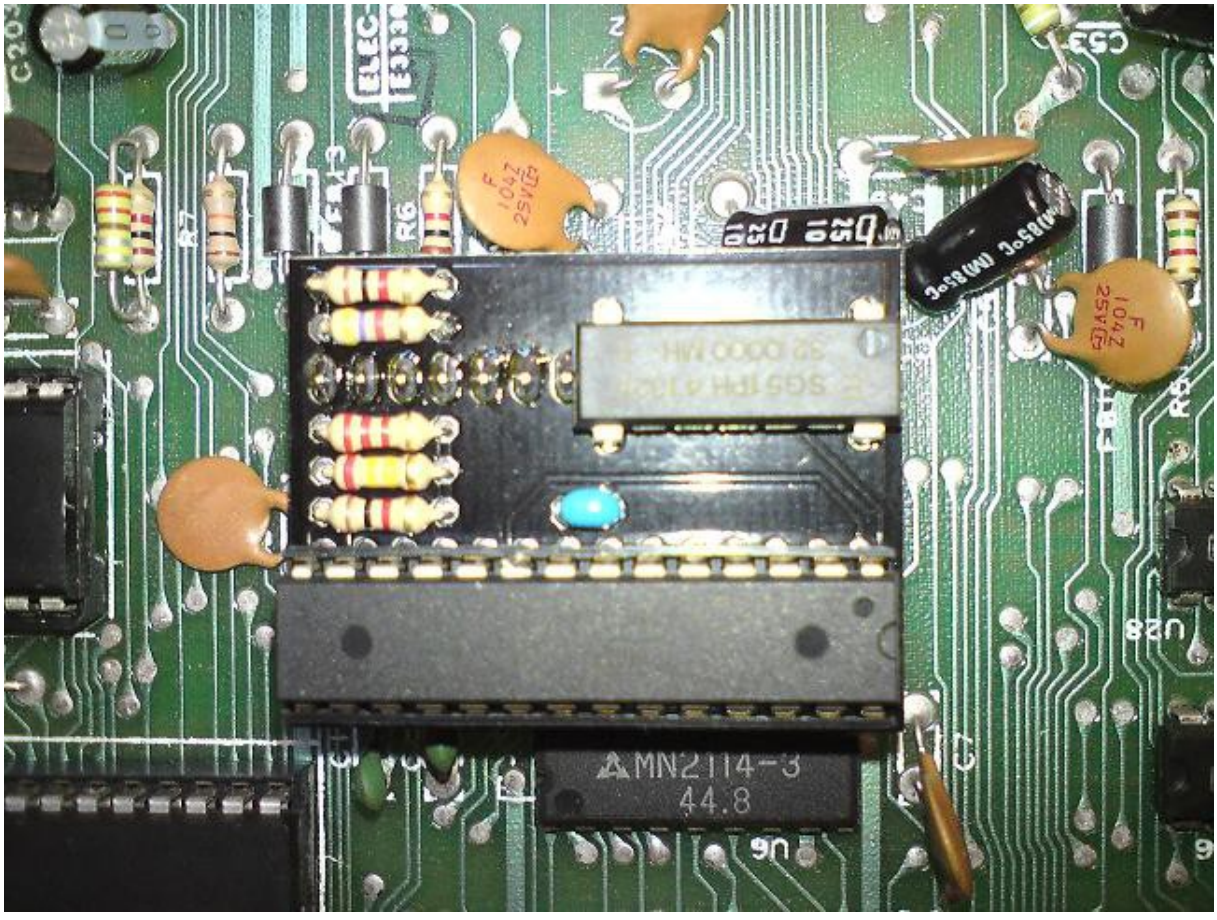
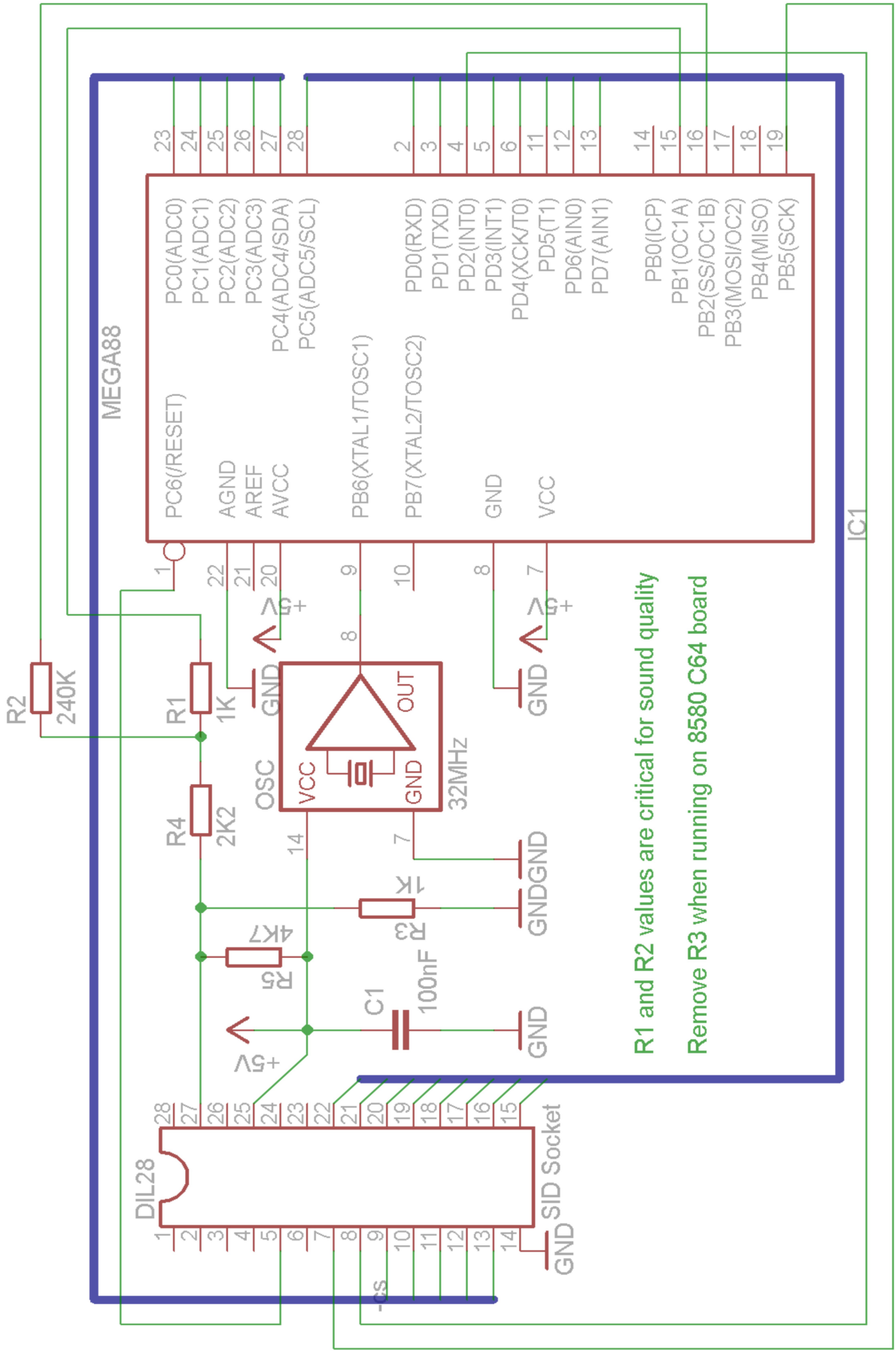


Bild des ersten Aufbaus mit der neuen 1.1 Platine:



Schaltbild:


MEGA88
IC1
PC6(/RESET)
PC0(ADC0)
PC1(ADC1)
PC2(ADC2)
PC3(ADC3)
PC4(ADC4/SDA)
PC5(ADC5/SCL)
AGND
AREF
AVCC
PB6(XTAL1/TOSC1)
PB7(XTAL2/TOSC2)
GND
VCC
PD0(RXD)
PD1(TXD)
PD2(INT0)
PD3(INT1)
PD4(XCK/T0)
PD5(T1)
PD6(AIN0)
PD7(AIN1)
PB0(ICP)
PB1(OC1A)
PB2(SS/OC1B)
PB3(MOSI/OC2)
PB4(MISO)
PB5(SCK)
R2
240K
R1
1K
R4
2K2
R3
1K
R5
4K7
C1
100nF
OSC
VCC
GND
OUT
32MHz
+5V
GND
GNDGND
DIL28
SID Socket
-cs
R1 and R2 values are critical for sound quality
Remove R3 when running on 8580 C64 board

Tiny Eprommer Nachbau

Von: Björn 'JMP$FCE2' Wieck

Eprom-Brenner für den C64 gibt es ja haufenweise und eigentlich tun sie recht zuverlässig wenn sie nicht gerade kaputt sind.

Bei den Alleskönnern aus dem Hause REX oder JANN bekommt man dann Probleme weil die auch Spezialchips haben die nicht mehr zu bekommen sind.

Der Tiny-Eprommer aus der 64´er 1987 ist da eher angenehm simpel weil komplett aus Standardteilen die auch noch in 10 Jahren beschaffbar sein sollten.

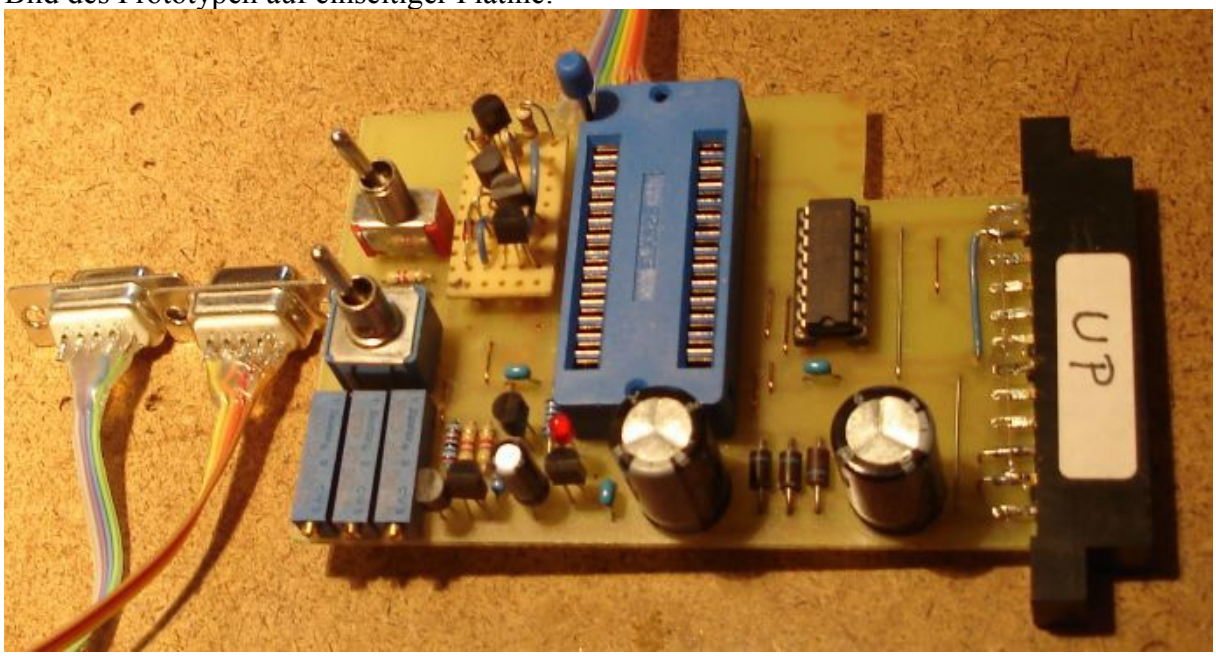
Leider hat auch der Tiny-Eprommer gewisse Probleme die sich aus der Einfachheit ergeben, so ist z.B. die Brennspannungsregelung über Z-Dioden eher ungenau und einige Epromtypen lassen sich damit nicht brennen. Desweiteren können damit keine 64kb (27512) Eproms gebrannt werden, was aber damals über eine Zwischenplatine realisiert wurde.

Das hat mich dann dazu gebracht hat den Tiny-Eprommer zu aktualisieren und einige Mängel zu beheben.

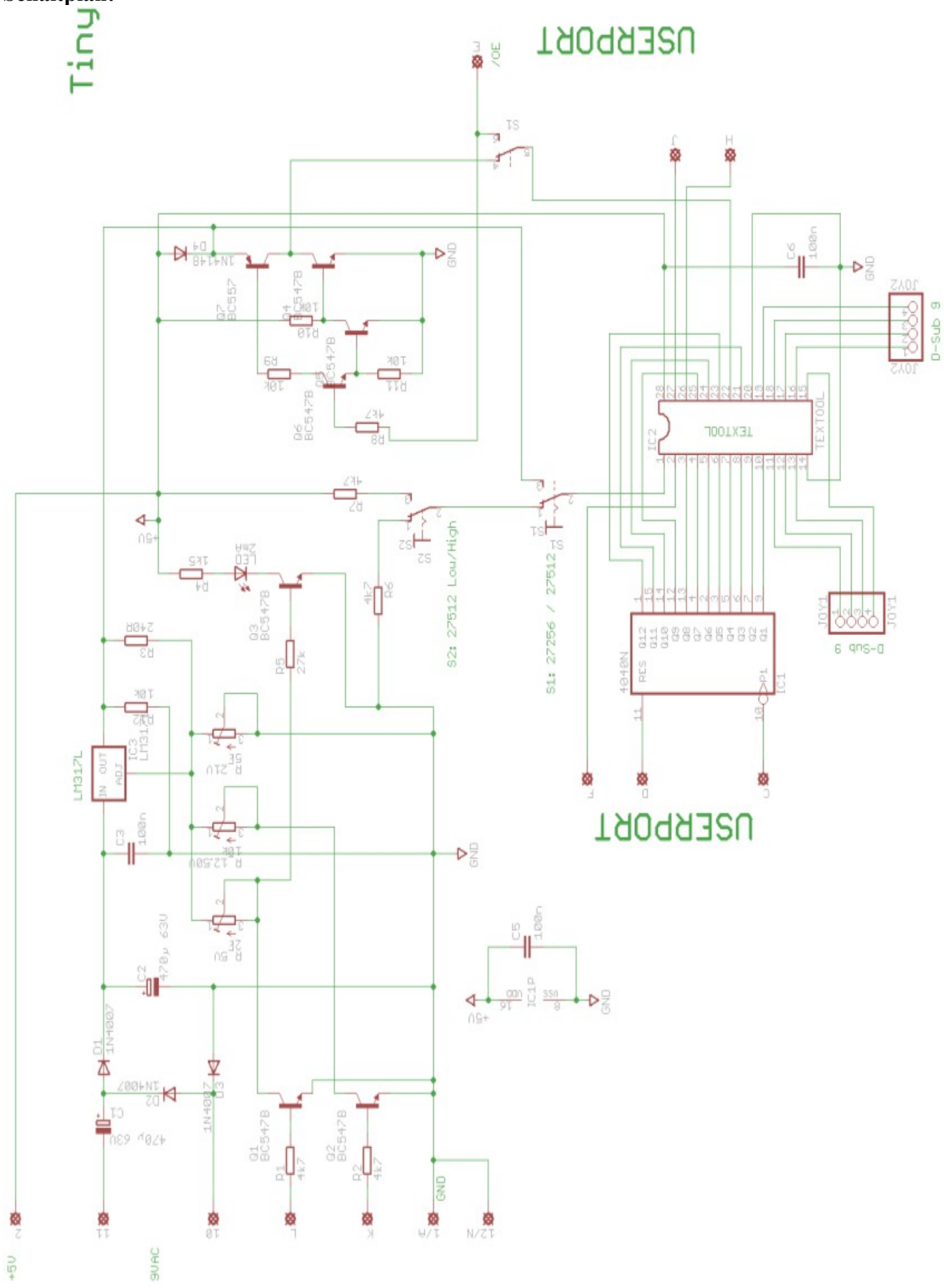
Die Brennspannungen sind nun justierbar und damit die Z-Dioden Exemplarstreuungen obsolet. 64kb (27512) lassen sich direkt brennen. Alle Bauteile sind wie vormals Standardtypen und daher leicht beschaffbar.

Der neue Tiny Eprommer hat zwei Schalter: der eine schaltet zwischen Normalbetrieb und 27512, der andere für den 27512 zwischen LOW und HIGH Bank

Bild des Prototypen auf einseitiger Platine:

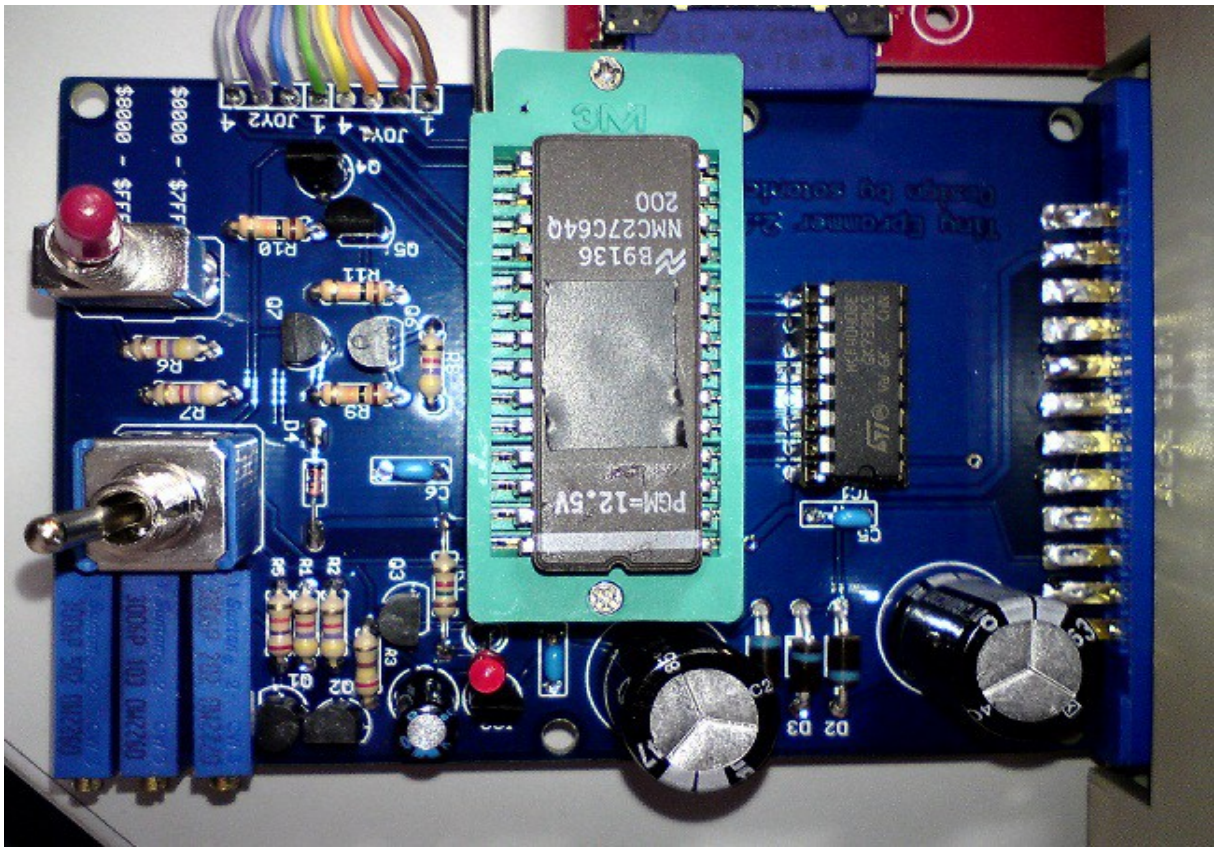


Schaltplan:

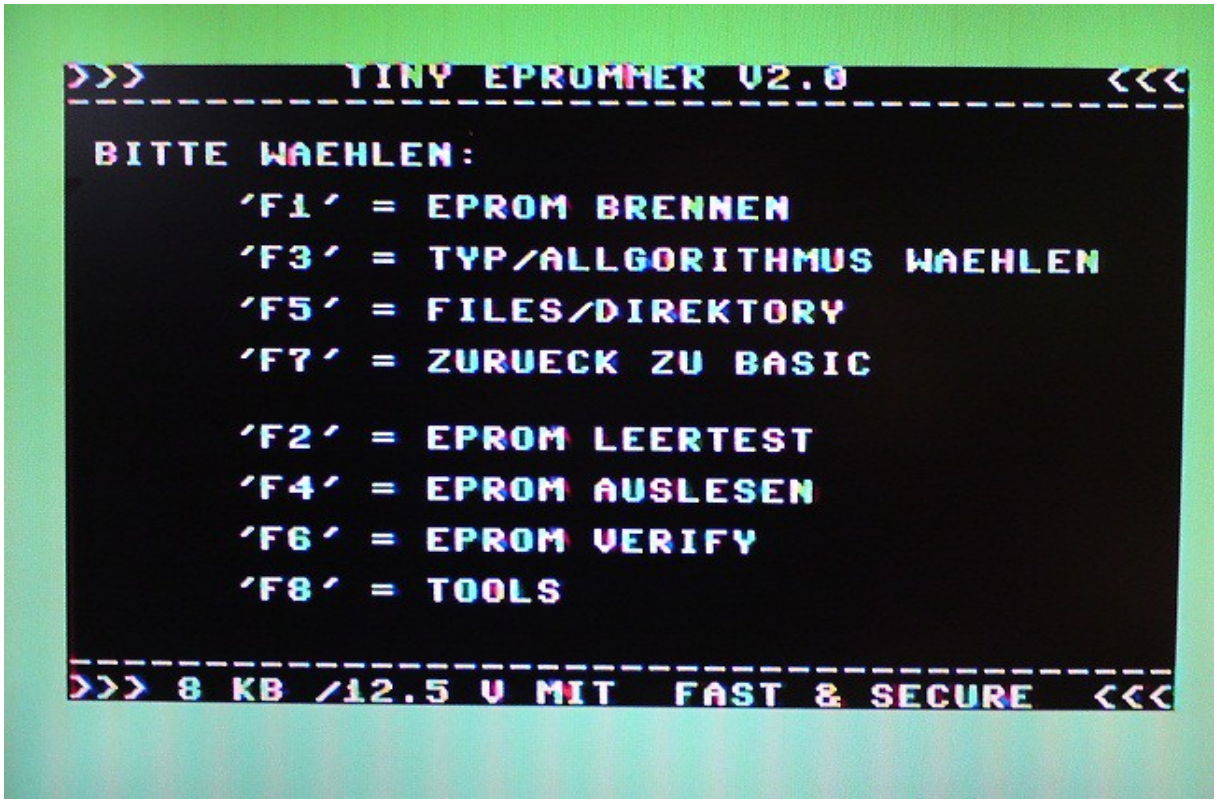

Tiny
USERPORT
USERPORT
S2: 27512 Low/High
S1: 27256 / 27512
TEXTOOL
D-Sub 9
D-Sub 9
LM317L
4040N
+5V
9VAC
GND

Sotonic aus dem Forum64 hat das einseitige Layout dann für eine Sammelbestellung in ein 2-seitiges Layout umgewandelt.

Bild des Aufbaus:



Bildschirmfoto:



Stückliste:

Bauteil	Wert	Device	Package	Reichelt Bestell Nr.
C1	470µ 63V	CPOL-EUE5-13	E5-13	RAD 470/63
C2	470µ 63V	CPOL-EUE5-13	E5-13	RAD 470/63
C3	100n	C-EU050-025X075	C050-025X075	KERKO 100N
C4	1µF	CPOL-EUE5-6	E5-6	RAD 1/63
C5	100n	C-EU050-025X075	C050-025X075	KERKO 100N
C6	100n	C-EU050-025X075	C050-025X075	KERKO 100N
D1	1N4007	1N4004	DO41-10	1N 4007
D2	1N4007	1N4004	DO41-10	1N 4007
D3	1N4007	1N4004	DO41-10	1N 4007
D4	1N4148	1N4148	DO35-10	1N 4148
IC1	4040N	4040N	DIL16	MOS 4040
IC2	TEXTOOL	DIL28	DIL28-6	TEX 28
IC3	LM317L	LM317L	317L	LM 317 TO 92
LED	2mA	LED3MM	LED3MM	LED 3MM 2MA RT
Q1	BC547B	BC547B	TO92	BC 547C
Q2	BC547B	BC547B	TO92	BC 547C
Q3	BC547B	BC547B	TO92	BC 547C
Q4	BC547B	BC547B	TO92	BC 547C
Q5	BC547B	BC547B	TO92	BC 547C
Q6	BC547B	BC547B	TO92	BC 547C
Q7	BC557	BC557	TO92	BC 557C
R1	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R2	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R3	240R	R-EU_0207/7	0207/7	METALL 240
R4	1k5	R-EU_0207/12	0207/12	METALL 1,50K
R5	27k	R-EU_0207/7	0207/7	METALL 27,0K
R6	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R7	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R8	4k7	R-EU_0207/7	0207/7	METALL 4,70K
R9	10k	R-EU_0207/7	0207/7	METALL 10,0K
R10	10k	R-EU_0207/7	0207/7	METALL 10,0K
R11	10k	R-EU_0207/7	0207/7	METALL 10,0K
R_5V	2K Lin	R-TRIMMT18	RTRIMT18	962-20 2,0K
R_12.75V	10k Lin	R-TRIMMT18	RTRIMT18	962-20 10K
R_21V	5k Lin	R-TRIMMT18	RTRIMT18	962-20 5,0K
S1		M9040P2	M9040P2	AS 500FPC
S2		M9040P	M9040P	AS 500APC
JOY1				D-SUB BU 09
JOY2				D-SUB BU 09

Kalibrieranleitung:
Schalter S1 und S2 nach OBEN schalten.
Brenner mit dem Userport verbinden, die Joystickstecker müssen nicht angeschlossen sein.
Multimeter mit dem TexTool-Sockel verbinden, Plus an Pin 1 und Minus an Pin 14.
C64 einschalten, das Multimeter sollte jetzt etwa 4,5V anzeigen.

POKE 56579,255
POKE 56577,0

Das Multimeter zeigt jetzt irgendwas zwischen 1,25 und 23 Volt an

Die Brennspannung ist jetzt eingeschaltet, die LED sollte leuchten,
mit dem 21V-Trimmer jetzt die Brennspannung auf genau 21 Volt einstellen.

mit

POKE56577,64

ist die Brennspannung für 12.5V aktiv und kann mit dem mittleren
Trimmer auf genau 12,5 Volt eingestellt werden.
(im Schaltplan steht 12.75V, das ist noch vom Betatest... 12,5V sind richtig)

zum Schluss noch

POKE 56577,128

Jetzt mit dem 5V-Trimmer genau auf 5 Volt einstellen.
Die LED sollte jetzt aus sein.

Test der 64kb Funktion:

POKE 56579,255
POKE56577,128

Multimeter mit dem TexTool-Sockel verbinden, Plus an Pin 1 und Minus an Pin 14.
Schalter S1 und S2 nach UNTEN schalten, das Multimeter zeigt jetzt 0 Volt an.
Schalter S2 nach OBEN schalten, das Multimeter zeigt jetzt ca. 5 Volt an.

Jetzt den Pluspol des Multimeters mit Pin 22 verbinden
Das Multimeter zeigt 0 Volt an.

POKE 56577,132 > Multimeter zeigt 5 Volt an, LED ist aus.

POKE 56577,68 > Multimeter zeigt 12,5 Volt an und die LED leuchtet.

POKE 56577,4 > Multimeter zeigt 21 Volt an und die LED leuchtet.

Damit ist der Test und Abgleich abgeschlossen.

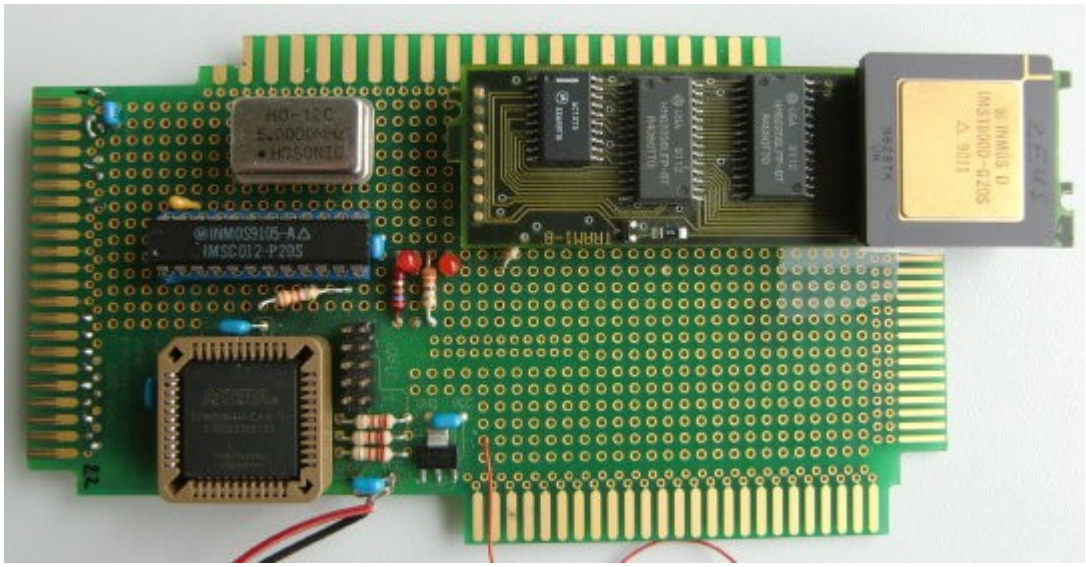
T2C64 – C64 und Transputer

Von Axel Muhr

Link: http://www.geekdot.com/index.php?page=transputer-to-8bit

Das dürfte eher ein Projekt für den extrem ‚technical mindet' User sein. Die zu erwartenden Kosten sollten sich eher im Bereich jenseits der 200€ Grenze einfinden und die Teile sind nicht leicht zu beschaffen. Trotzdem möchte ich hier zeigen, das so etwas geht und interessante technische Aspekte eröffnet.

Über ein 8-bit Baby, CPLD, C012/1 Link Interface Chip wird ein Transputer (TRAM Karte) an einen C64 angebunden:



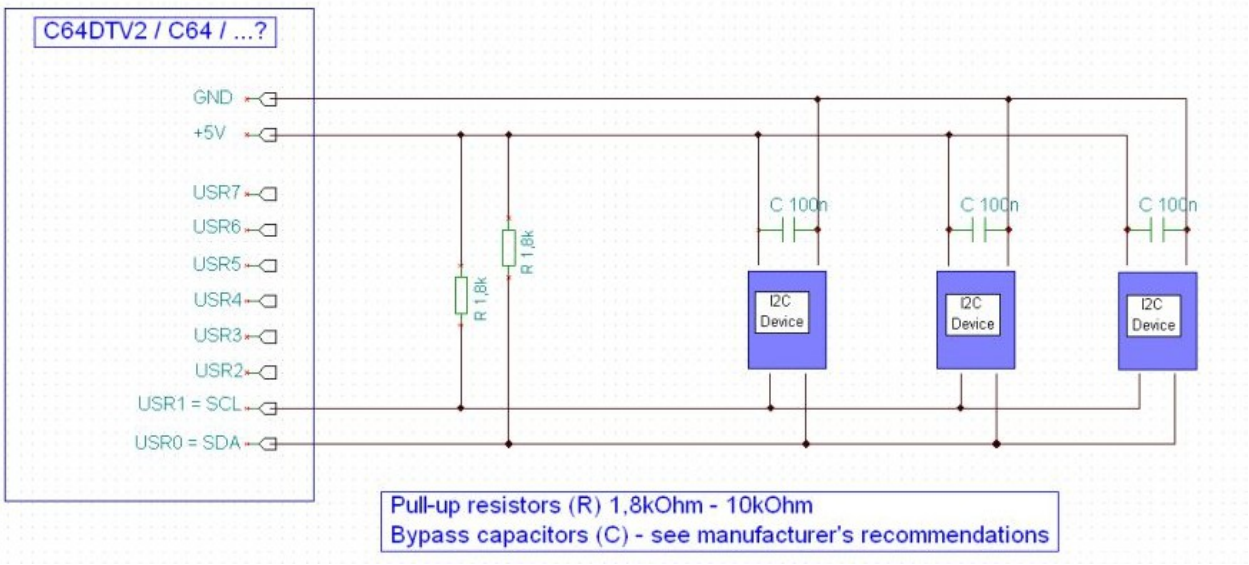
In einem ersten Demoprogramm wird die Punktberechnung für eine Mandelbrotgraphik in der Auflösung 320x200x8 an den Transputer übergeben (32 Iterationen; Floatingpoint mit doppelter Genauigkeit):

I2C Projekte

Link:
https://sites.google.com/site/dividedbit/home/c64-projects

Timo Voutilainen hat mit I2C am C64 experimentiert und herausgekommen sind interessante Projekte. Die Software ist in C geschrieben und mit dem CC65 kompiliert. I2C ist ein von Philips Semiconductors (heute NXP Semiconductors) entwickelter serieller Datenbus zur Chip <-> Chip Kommunikation. I2C ist technisch identisch mit TWI (Two Wire Bus).

Die I2C Bausteine werden zumeist wie folgt an den C64 angeschlossen:



I2Cpeek

Hier lassen sich I2C Bausteine ansprechen und austesten.

FM Radio

Hier wird ein RDA5807 Radio Modul verwendet:

Die Verbindungen zum Commodore 64 User port:

C64 Pin	Pin Name	RDA5807 Module pin
1	GND	GND
2	+5V	VCC
C	PB0	DAT
D	PB1	CLK

Bildschirm des laufenden C64 Steuerprogramms:

VC20 Multi Cartridge

Ein defektes VC20 Modul kann man sehr einfach mittels eines größeren Eproms in ein Mehrfach-Spielemodul umbauen. Es wird jeweils ein 8k Rombereich auf $A000 über BLK 5 Signal und Auswahl über die Adressleitungen A13.. selektiert.

VC20 Expansionsport (Von Rückseite gesehen):

```
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |


|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
A  B  C  D  E  F  H  J  K  L  M  N  P  R  S  T  U  V  W  X  Y  Z
```

1 - GND	A - GND
2 - CD0	B - CA0
3 - CD1	C - CA1
4 - CD2	D - CA2
5 - CD3	E - CA3
6 - CD4	F - CA4
7 - CD5	H - CA5
8 - CD6	J - CA6
9 - CD7	K - CA7
10 - !BLK1	L - CA8
11 - !BLK2	M - CA9
12 - !BLK3	N - CA10
13 - !BLK5	P - CA11
14 - !RAM1	R - CA12
15 - !RAM2	S - CA13
16 - !RAM3	T - I/O2
17 - VR/W	U - I/O3
18 - CR/W	V - SO2
19 - !IRQ	W - !NMI
20 - NC	X - !RESET
21 - +5V	Y - NC
22 - GND	Z - GND

27C256 Eprom und Beschaltung:

```
          27256
    +-------\/-------+
 1 -|Vpp          Vcc|- 28
 2 -|A12          A14|- 27
 3 -|A7           A13|- 26
 4 -|A6            A8|- 25
 5 -|A5            A9|- 24
 6 -|A4           A11|- 23
 7 -|A3           /OE|- 22
 8 -|A2           A10|- 21
 9 -|A1      /CE /pgm|- 20
10 -|A0            D7|- 19
11 -|D0            D6|- 18
12 -|D1            D5|- 17
13 -|D2            D4|- 16
14 -|gnd           D3|- 15
    +----------------+
```

Alte IC's auslöten uns sockeln.
+5V und Gnd sollte klar sein.
D0-D7 auch; gehen direkt ans Eprom.
A0-12 auch; gehen auch ans Eprom.
VPP an +5V legen.
/CE an BLK5 legen.

Bild von oben:

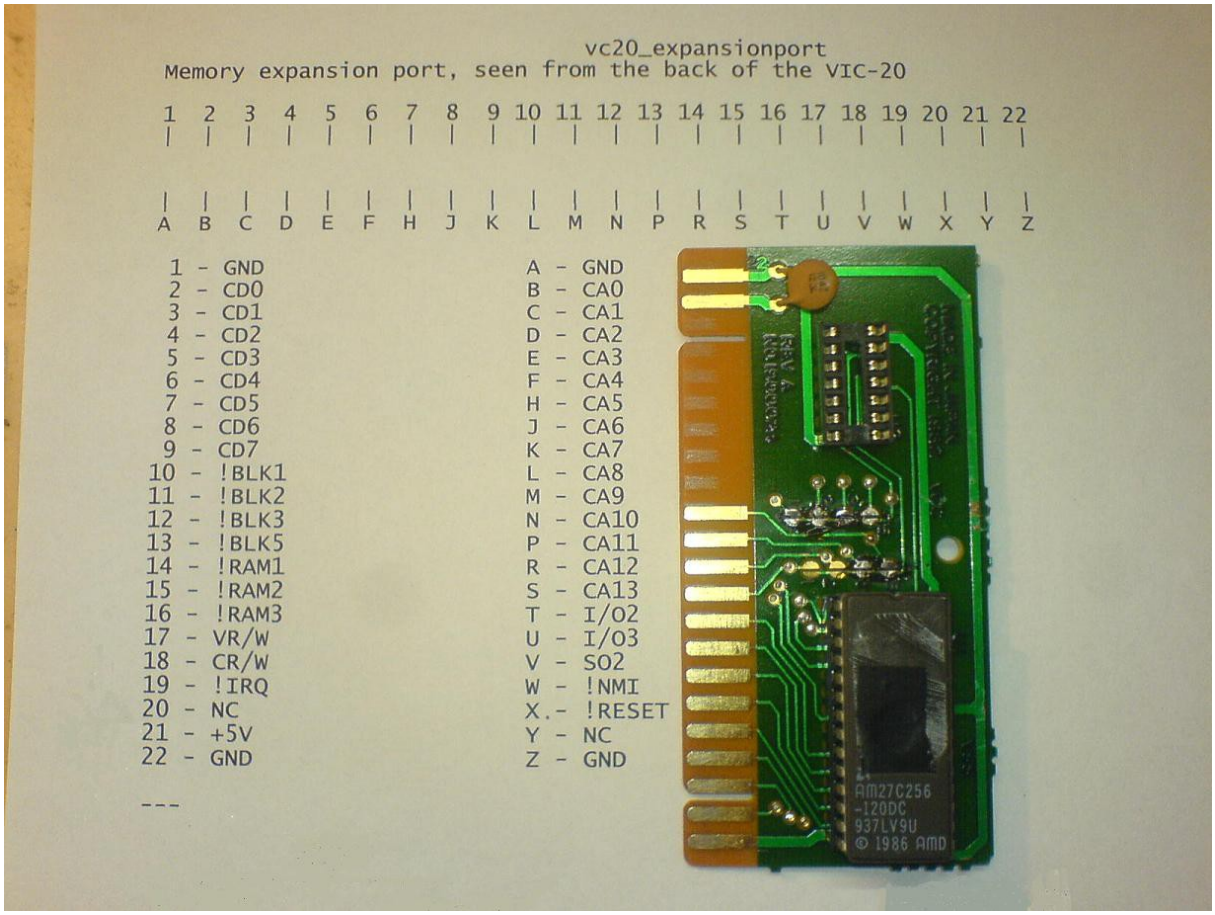
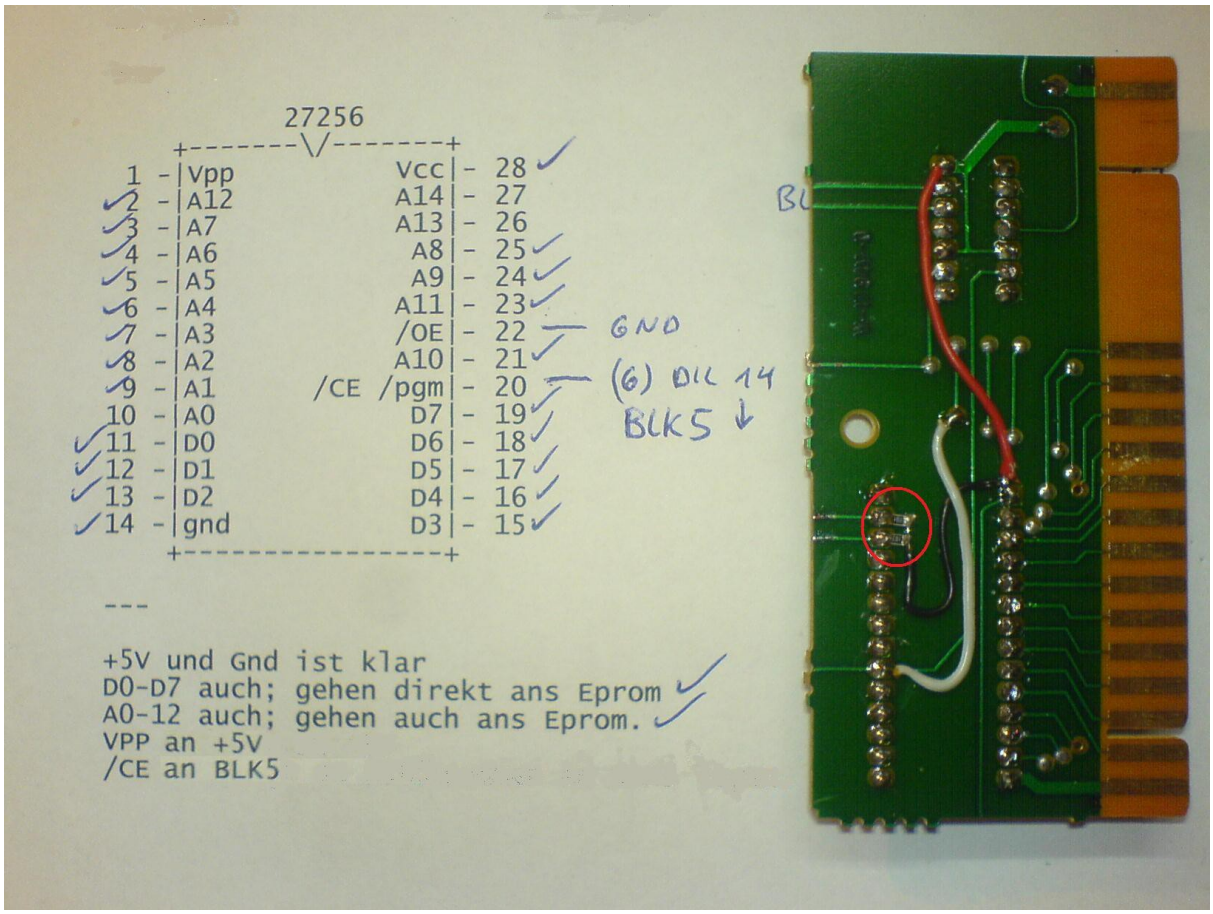


Bild von unten:



Im rot markierten Bereich sind die A13 und A14 Adressleitungen über 10k SMD Widerstände an +5V gelegt (Pullup). Über Jumper/Schalter an GND lassen sich so die 4 x 8k Rombereich anwählen.

Final Expansion für VC20

Nach Ing. Thomas Winkler

Der VC20 war einer der ersten Homecomputer der in Masse produziert wurde. Keiner glaubte wirklich an einen Erfolg, Commodore produzierte ihn hauptsächlich weil zuviele RAM Chips auf Lager waren. Zudem steckte man viel Geld in die Entwicklung eines Grafikchip (VIC), aber keine Firma wollte ihn kaufen. Doch sehr schnell wurde der VC20 ein großer Markterfolg und bereitete den Weg für den C64 ...

Im Gegensatz zum C64 hat der VC20 nur sehr wenig RAM, nur 5KB und davon stehen nur 3,5KB zur freien Verfügung. Deswegen wurden auch bald Speichererweiterungen verkauft. Erst eine 3KB Erweiterung, dann eine 8KB und eine 16KB. Man kann die Erweiterungen teilweise auch kombinieren, weil sie für unterschiedliche Speicherbereiche konzipiert sind. Es kamen einige Spiele und Programme für den VC20 auf den Markt, und leider benötigen viele der Spiele und Programme unterschiedliche Speichererweiterungen. Will man jede Software laufen lassen, so benötigt man alle Speichererweiterungen. Einige Programme laufen sogar nicht, wenn zuviel Speicher vorhanden ist.

Viele Spiele und Utilities gibt es in Form einer ROM Erweiterung (Cartridge). Manche belegen 8KB Speicher, andere 16KB. Einige dieser Cartridges laufen nur in einem ROM, weil sie in den eigenen Speicher schreiben. Ladet man diese Programme in eine Speichererweiterung, dann zerstören sie sich selbst.

Der VC20 kann wie der C64 exakt 64KB adressieren. Davon sind 16KB belegt durch Kernel (8KB) und dem Basic Interpreter (8KB). Weiter 8KB sind für IO Ressourcen reserviert. Die restlichen 40KB sind frei oder mit internem RAM belegt.

Aufgrund der Problematik der verschiedensten Speicherkonfiguration wurde die Idee des Final Expansion geboren.

Der erste Prototyp der **Final Expansion** war handgefädelt und hatte 2 SRAM Bausteine mit insgesamt 40KB. Der zweite Prototyp der **Final Expansion** ist auf selbstgefertigter 2-fach Layer Printplatte (Dank an **for(;;)**) aufgebaut. Vor der Produktion einer Print kam im Forum 64 die Idee, die FE mit der genialen SD2IEC Platine zu kombinieren. Damit hat man zusätzlich Zugriff auf bis zu 2GB in Form einer modernen SD Karte und kann auf eine Floppy vollends verzichten. Dank **JMP$FCE2** wurde das extrem dichte Layout auf engsten Raum möglich, und das mit nur 2-fach Layer. Nun steht eine moderne Cartridge für den VC-20 zur Verfügung:

- Erweiterung auf maximal 512KB RAM (max. 40KB sichtbar, rest in Banks). Dabei sollen alle gängigen RAM Ausbauten einstellbar sein: 3KB, 8KB, 11KB, 16KB, 19KB, 24KB, 27KB, (35KB)
- Jeder einzelne Speicherblock läßt sich gegen überschreiben schützen, sodass er wie ein ROM aussieht (ladbare Cartridge).
- Jeder einzelne Speicherblock läßt sich deaktivieren, sodass die FE mit anderen Cartridges betrieben werden kann.
- Eine EEPROM Erweiterung um 512KB . Das EEPROM ist vom VC20 aus programmierbar.

- Bis zu 2 GB Massenspeicher in Form einer modernen SD Karte als zuverlässiger Floppy Ersatz. Dadurch einfacher Datenaustausch mit dem PC (und damit dem Internet) plus Starten von Programmen und Modulimages (die keineswegs erst in das Flash geschrieben werden müssen, sondern auch direkt von der SD-Karte geladen werden können) Zudem kann es extern durch ein LCD Display erweitert werden. Ein eingebauter Schnelllader unterstützt die Turboroutinen im sd2iec.
- Zwei Register an einer IO Adresse zur Konfiguration der FE per Software. Die Register lassen sich ausblenden.
- Eine Batteriegepufferte Echtzeituhr.

Mein Dank geht an alle Personen die fleißig an dem Projekt mitarbeiten, insbesondere an **Peter Sieg**, **JMP$FCE2**, **for(;;), Wiesel, Unseen** und **Sign-Set** vom Forum 64.

Was kann man damit machen

Nach dem Einschalten des VC20 (bzw. nach einem Hardreset) ist ein 8KB grosses Stück des EEPROM der Final Expansion aktiviert (sichtbar). Das Kernel des VC20 findet eine Cartridge und startet die Firmware des Final Expansion. Wenn keine Taste gedrückt wird, dann kommt ein Menü mit dem man die Final Expansion konfigurieren kann:

- Speicherausbau wählen und VC20 Basic starten. Man wählt eine RAM Erweiterung und startet mit <Enter>.
- Eine Cartridge aus dem EEPROM direkt starten. Man wählt ein Spiel oder Programm und startet mit <Enter>.
- Ein Spiel oder Programm direkt von einer SD oder Diskette starten. Das Verzeichnis der Diskette wird angezeigt und man wählt einen Eintrag und startet mit <Enter>.

Mit Zusatzprogrammen kann man die Final Expansion nach eigenen Wünschen konfigurieren:

- Firmware Flasher: damit kann man die Firmware der FE in das EEPROM schreiben.
- Cartridge oder Programm in das EEPROM speichern für den Cartloader.

Das Konzept der LOADER-Dateien

Diese Dateien enthalten neben der im Flash abgelegten Firmware die "Intelligenz" des Moduls. Dem Benutzer präsentieren sie sich als selbst veränderbares "Startmenü", das dem Benutzer nur noch ausführbare Programme anzeigt und andere, für den Benutzer unwichtige Dateien verbirgt. In den LOADER-Dateien ist hinterlegt, ob das Programm ein BASIC-Programm, ein Cartridge oder ein Maschinencodeprogramm ist und welche Speicherkonfiguration das Programm benötigt. Einmal angelegt, muss sich dann der Benutzer um diese Details nicht mehr kümmern und sich auch nicht mehr merken, welches Programm welche Speicherkonfiguration benötigt.

Aufbau der Final Expansion v3.1

Es wird eine Printplatte gefertigt, die den Slot Anschluß für den VC20 hat und Platz für den CPLD (PLCC-44 Sockel) sowie den beiden Speicherbausteinen (DIL-32) bietet. Der CPLD ersetzt eine Menge TTL Chips und ist für die Adressdekodierung sowie für die Software gesteuerten Konfiguration (Latch) zuständig. Der CPLD muß erst per JTAG für seine Aufgabe programmiert werden. Atmel bietet hierzu kostenlos eine Software an (WinCUPL), mit der man die Logik des CPLD "verdrahten" kann.

Neben dem CPLD und ein paar Bauteile benötigt man noch den externen Speicher und einen Controller:

- ein 512KB SRAM (628512) als Erweiterung des Arbeitsspeicher.
- ein 512KB EEPROM (29F040) als Speicher für die Firmware (Modul Menüprogram) und oft benutzte Programme bzw. Cartridges.
- optional ein Atmega644 und eine SD Karte als Massenspeicher bzw. als Floppy Ersatz (SD2IEC).

Die Cartridge hat verschieden Betriebsmodi:

- Start Modus: Modus nach dem Einschalten bzw. nach einem Reset. Es ist nur ein 8KB großes Stück des EEPROM an der Adresse $A000 sichtbar, damit die Firmware startet.
- ROM Modus: Dieser Modus blendet das EEPROM ein. Das EEPROM ist in allen Speicherbereichen sichtbar (32KB zur selben Zeit). Schreibzugriffe gehen in das SRAM, so kann man einfach vom EEPROM in das SRAM kopieren.
- RAM Modus: Speichererweiterung mit programmierbarer Größe und optionalen Schreibschutz.
- Super RAM: In diesem Modus kann das ganze 512KB RAM in 32KB Stücke eingeblendet werden. Schreibschutz funktioniert hier nicht.
- OFF Modus: Cartridge ist "unsichtbar", also ausgeschaltet (Reaktivierung nur durch Reset).
- Program Modus: In dem Modus kann das EEPROM geflashed werden. Achtung nur aktivieren, wenn man weiss was man tut ...

Hauptmenue:

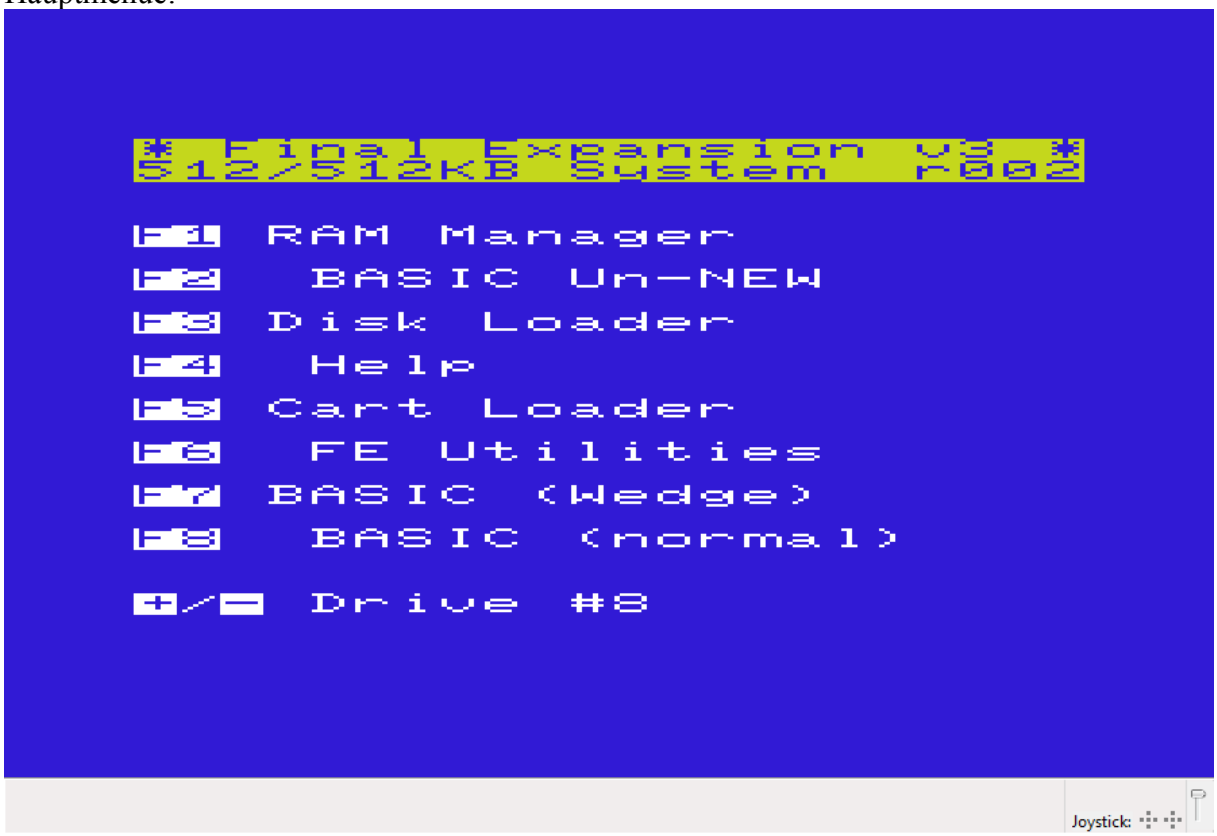
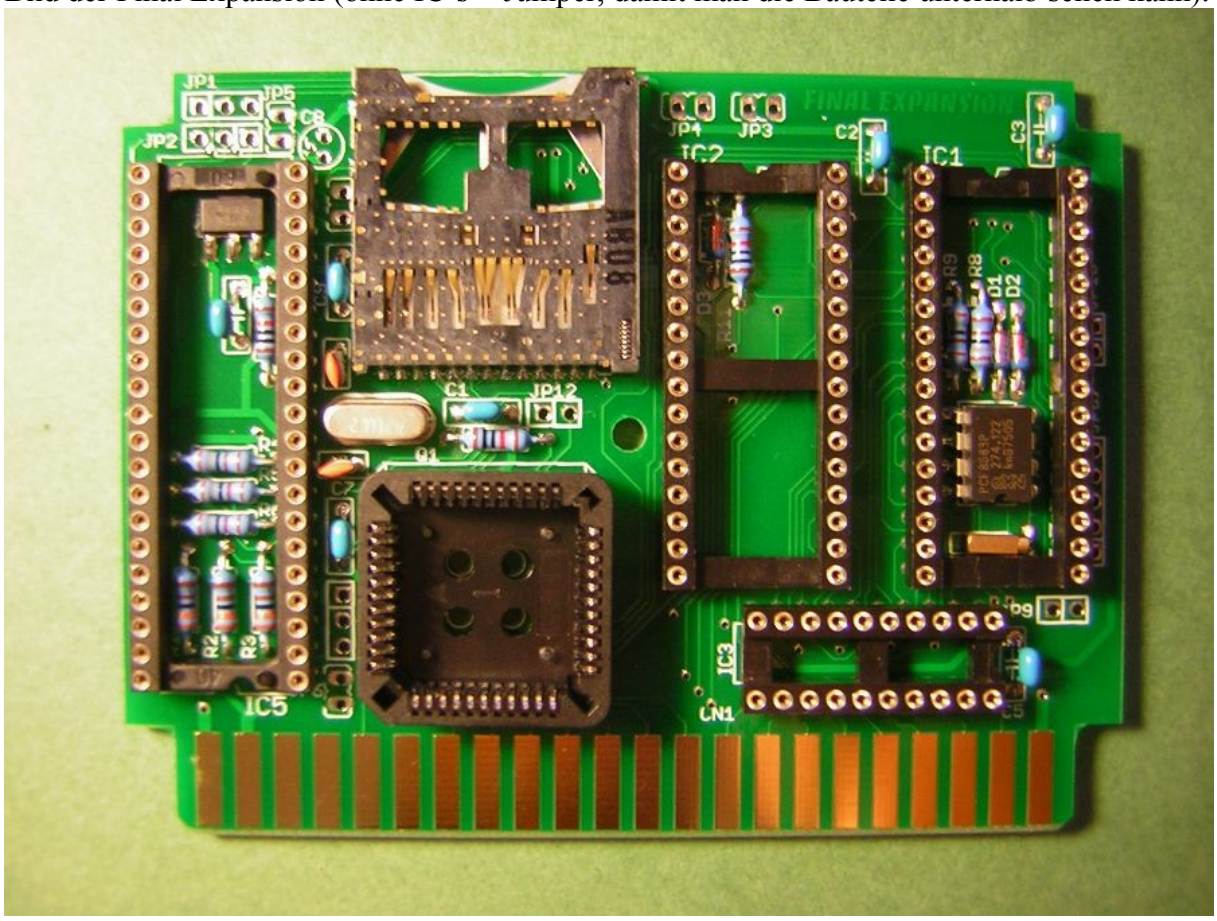
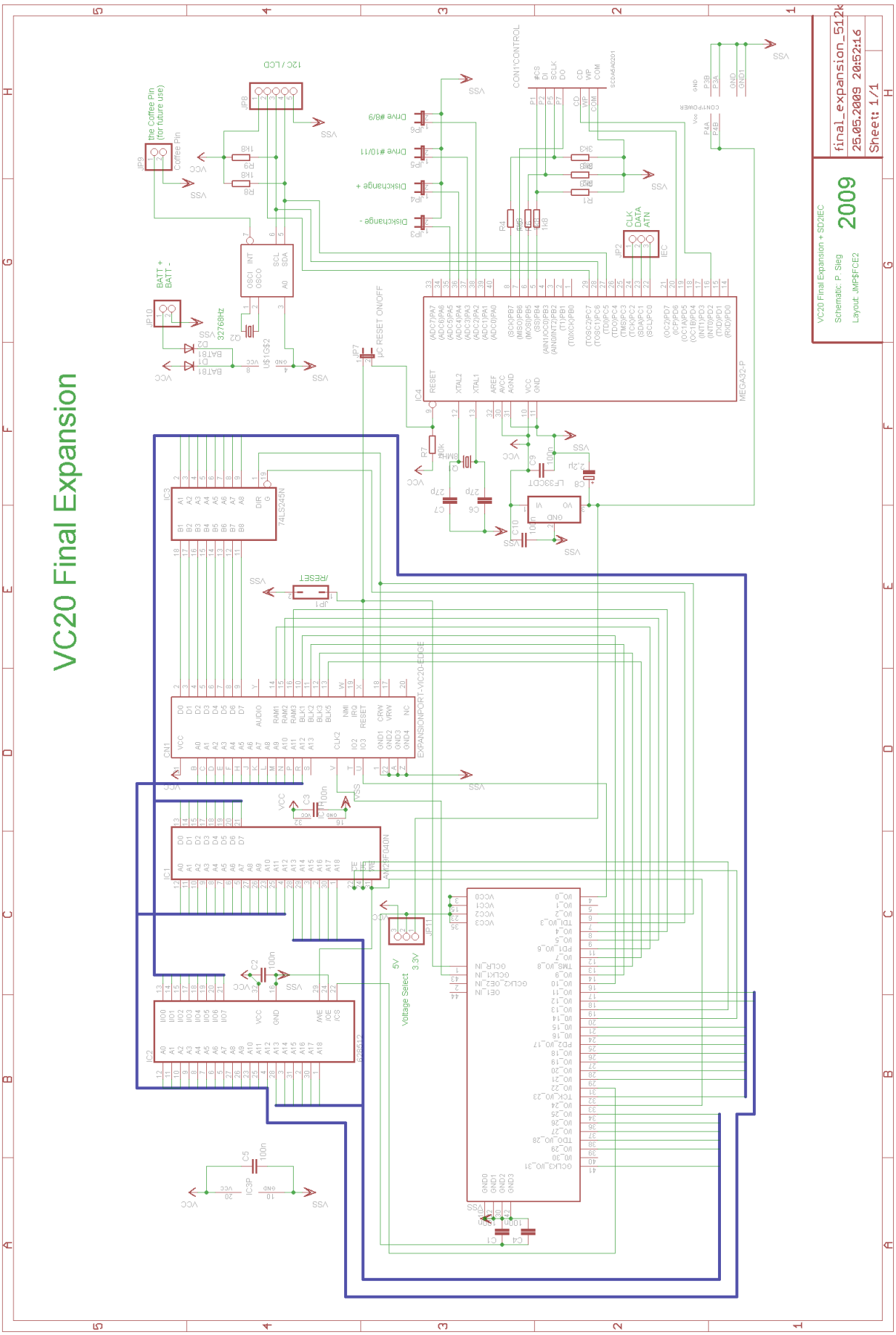

* Final Expansion V3 *
512/512KB System r002
F1 RAM Manager
F2 BASIC Un-NEW
F3 Disk Loader
F4 Help
F5 Cart Loader
F6 FE Utilities
F7 BASIC <Wedge>
F8 BASIC <normal>
+/- Drive #8
Joystick:

Bild der Final Expansion (ohne IC’s + Jumper, damit man die Bauteile unterhalb sehen kann):



Links zum Projekt:
http://vc20final.t-winkler.net/index.html
http://www.for8bits.com/

Schaltbild:


VC20 Final Expansion
final_expansion_512k
25.05.2009 20:52:16
Sheet: 1/1
VC20 Final Expansion + SD2IEC
Schematic: P. Sieg
Layout: JMP$FCE2
2009
MEGA32-P
EXPANSIONPORT-VIC20-EDGE
74LS245N
AM29F040N
Voltage Select
CON1'CONTROL
12C / LCD
Drive #8/9
Drive #10/11
Diskchange +
Diskchange -
the Coffee Pin (for future use)
Coffee Pin
BATT +
BATT -
32768Hz
µC RESET ON/OFF
LF33CDT
/RESET
CLK
DATA
ATN
IEC
CONTPOWER
VSS
VCC

Neigungssensor + AVR als Joystick

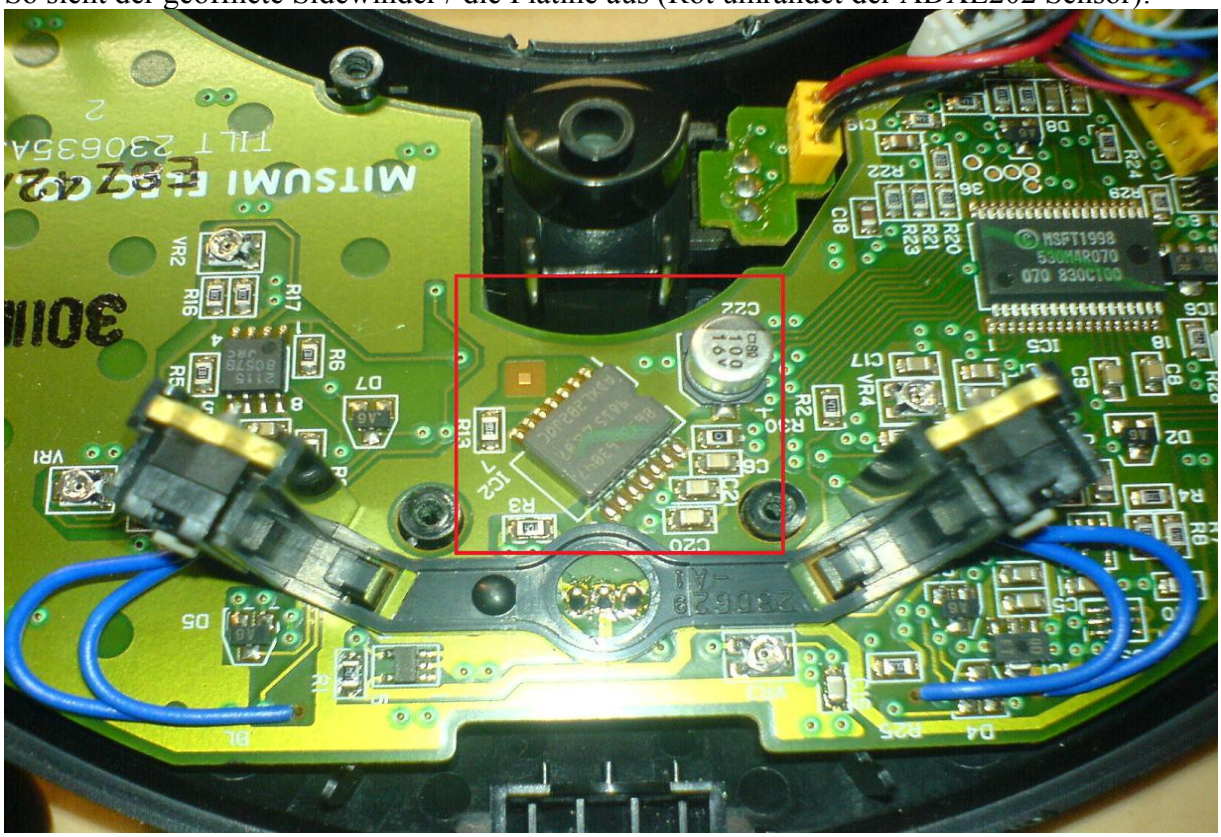
Nach Anregung von Till Harbaum: http://www.harbaum.org/till/palm

Die wunderbare Seite von Till ist immer wieder einen Besuch wert! Die Programme sind GPL.

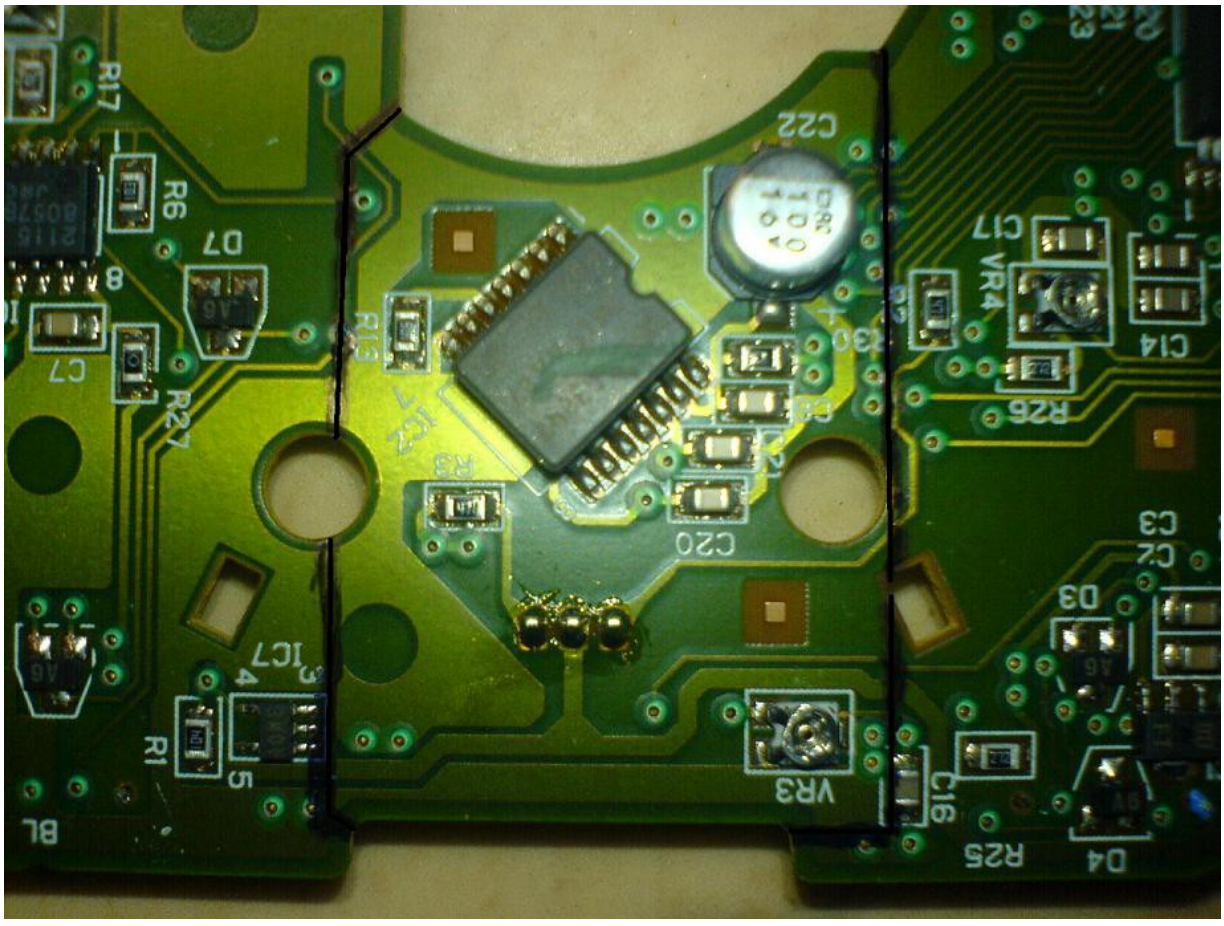
Ich hatte vor Jahren schon einmal diese tolle Seite besucht.. aber der verwendete Sensor ADXL202 war mit zu teuer. In 2009 kostet er bei segor ca. 26€.. Über Kongkong habe ich mir einen für ca. 20€ inkl. Versand bestellt. Dann habe ich durch weitere Internet Recherchen herausgefunden, das dieser Sensor auch im Microsoft Sidewinder Freestyle verbaut ist! Und da kommt man heute nicht nur günstig dran, sondern es ist auch schon fast alles aufgebaut!

So sieht der Microsoft Sidewinder Freestyle aus. Der mit dem "Bewegungssensor"!

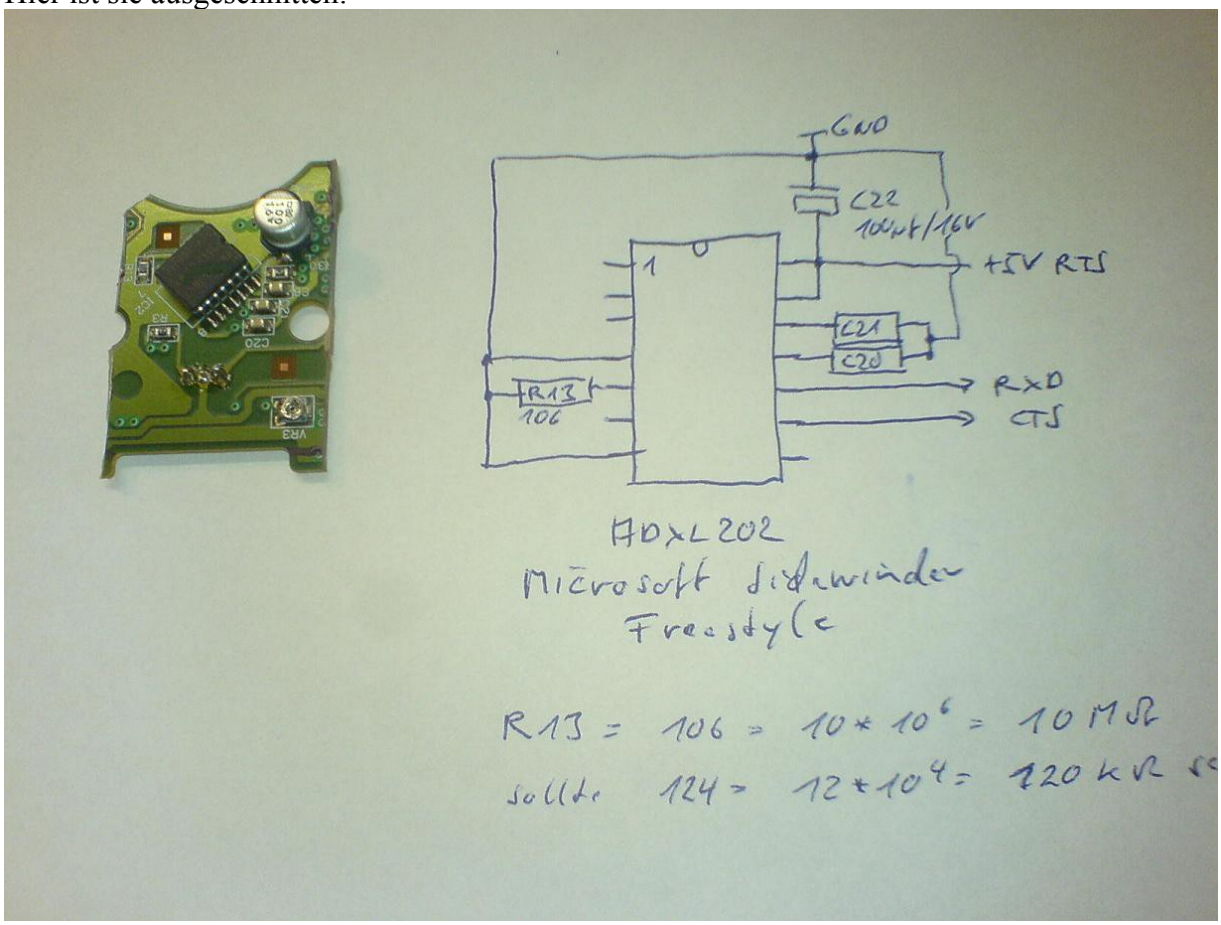
So sieht der geöffnete Sidewinder / die Platine aus (Rot umrandet der ADXL202 Sensor):



Hier sollte man die Platine trennen (Dremel). Die LED auf der anderen Seite einfach abkneifen/auslöten:

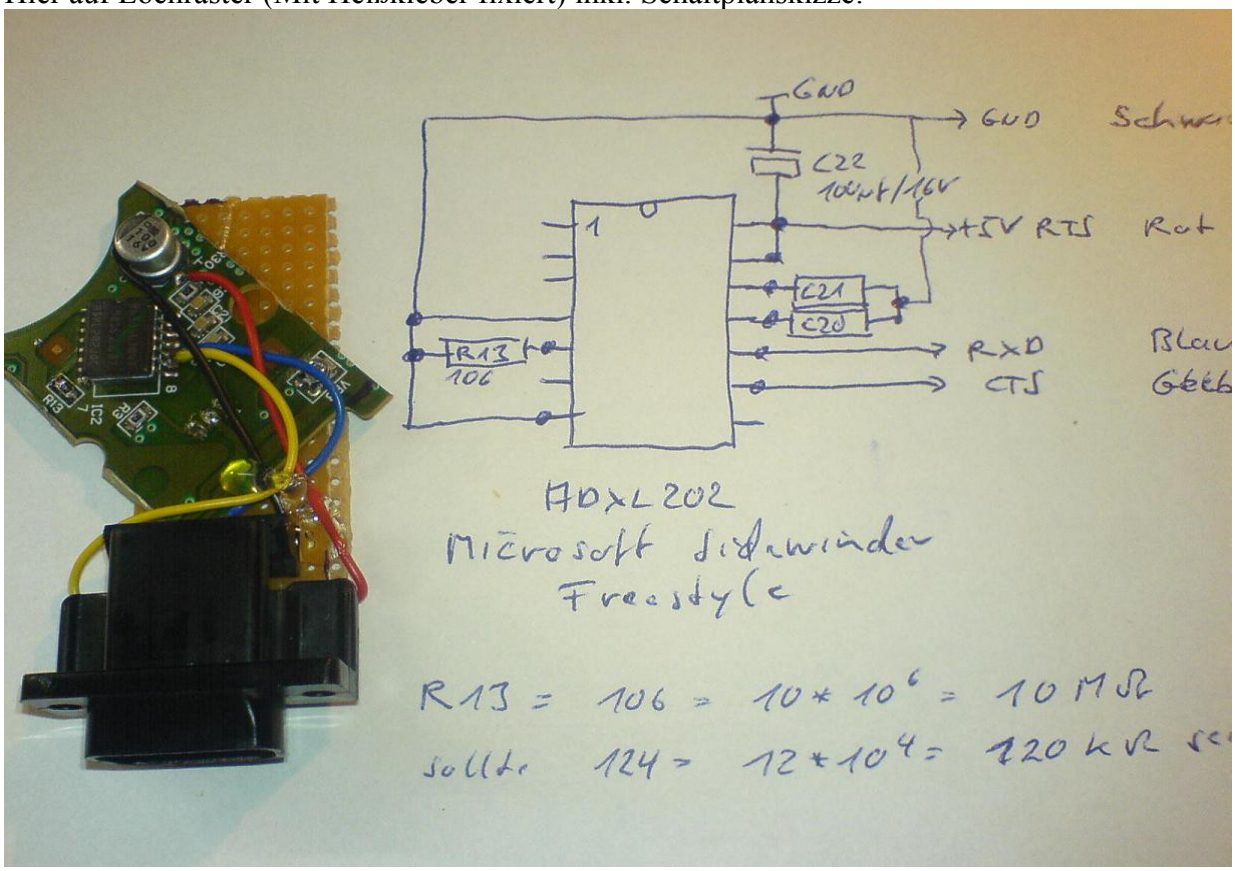


Hier ist sie ausgeschnitten:

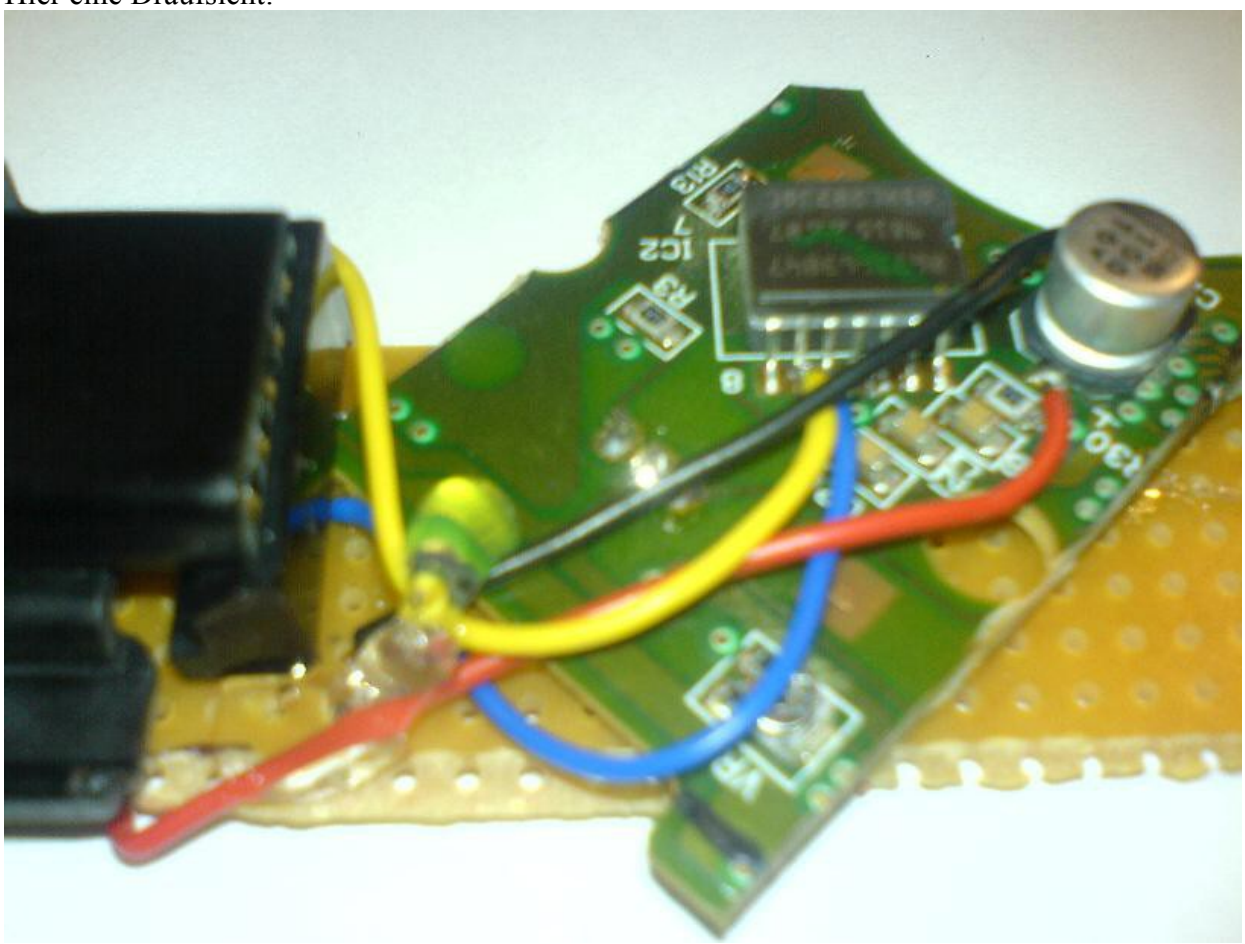


Per Multimeter und Nachmessen der Verbindungen, erkennt man, das eigendlich alles schon korrekt drauf ist. Nur der Widerstand R13 ist mit 106 = 10 * 10^6 = 10MOhm ca. 100x zu groß! Aber auf den restlichen Platinenteilen ist ein 104 = 10 * 10^4 = 100kOhm SMD Widerstand zu finden. Den 106 entfernen und den 104 anstatt dessen einlöten!
(Das übt man am besten zuerst an anderen SMD Teilen auf den Reststücken. Einfach beide Seiten mit dem Lötkolben erhitzen, bis der Widerstand sich von den Lötpads einfach wegschieben läßt. Sollte er am Lötkolben kleben, einfach mit einer Pinzette abnehmen.
Es müssen dann auf einer Lochrasterplatine als Träger, ein Sub-D Stecker eingelötet und 4 dünne Kabel zur Sidewinder Platine verbunden werden (+5V und GND zum 100uF Kondensator, RTS+CTS zum ADXL202).

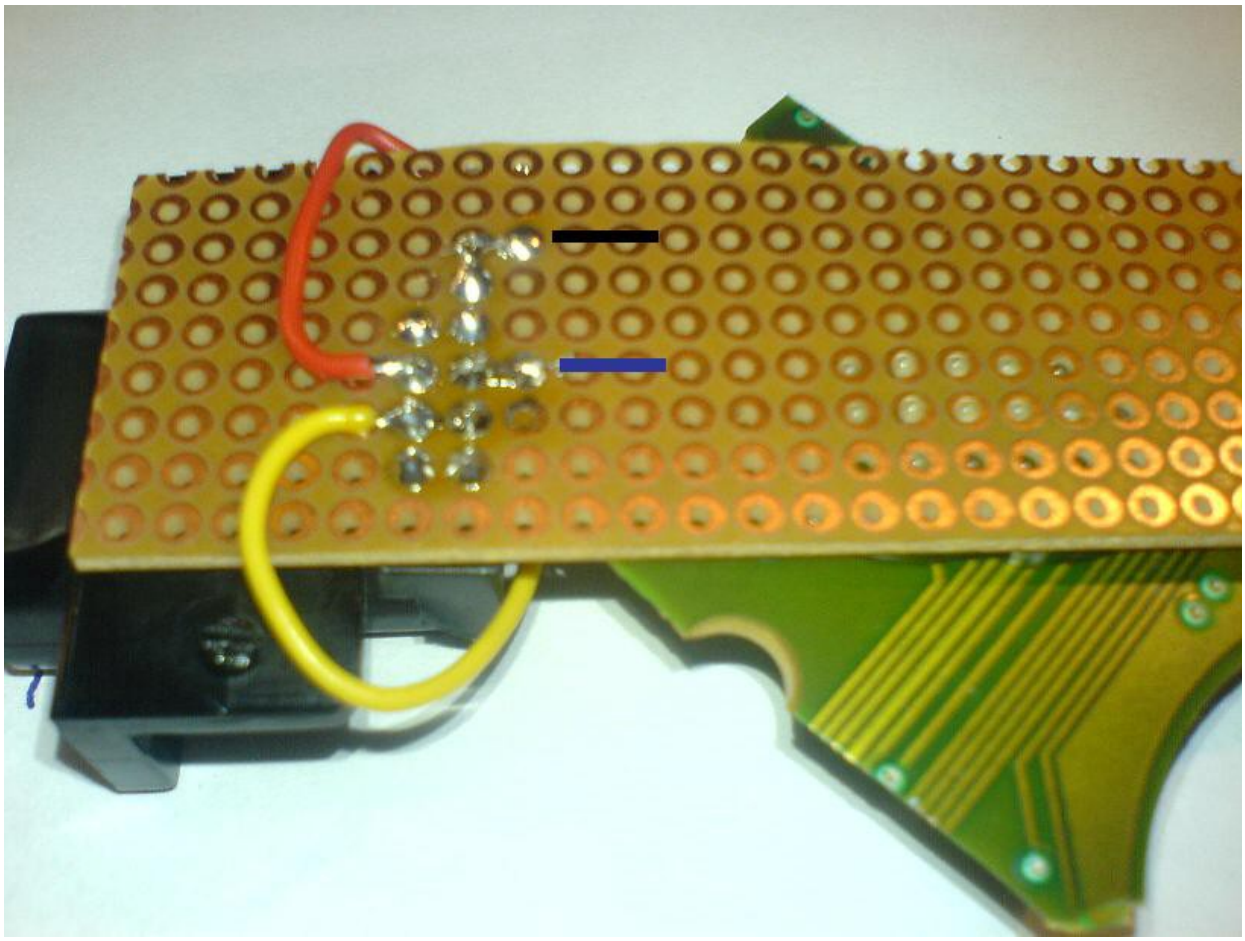
Hier auf Lochraster (Mit Heißkleber fixiert) inkl. Schaltplanskizze:



Hier eine Draufsicht:

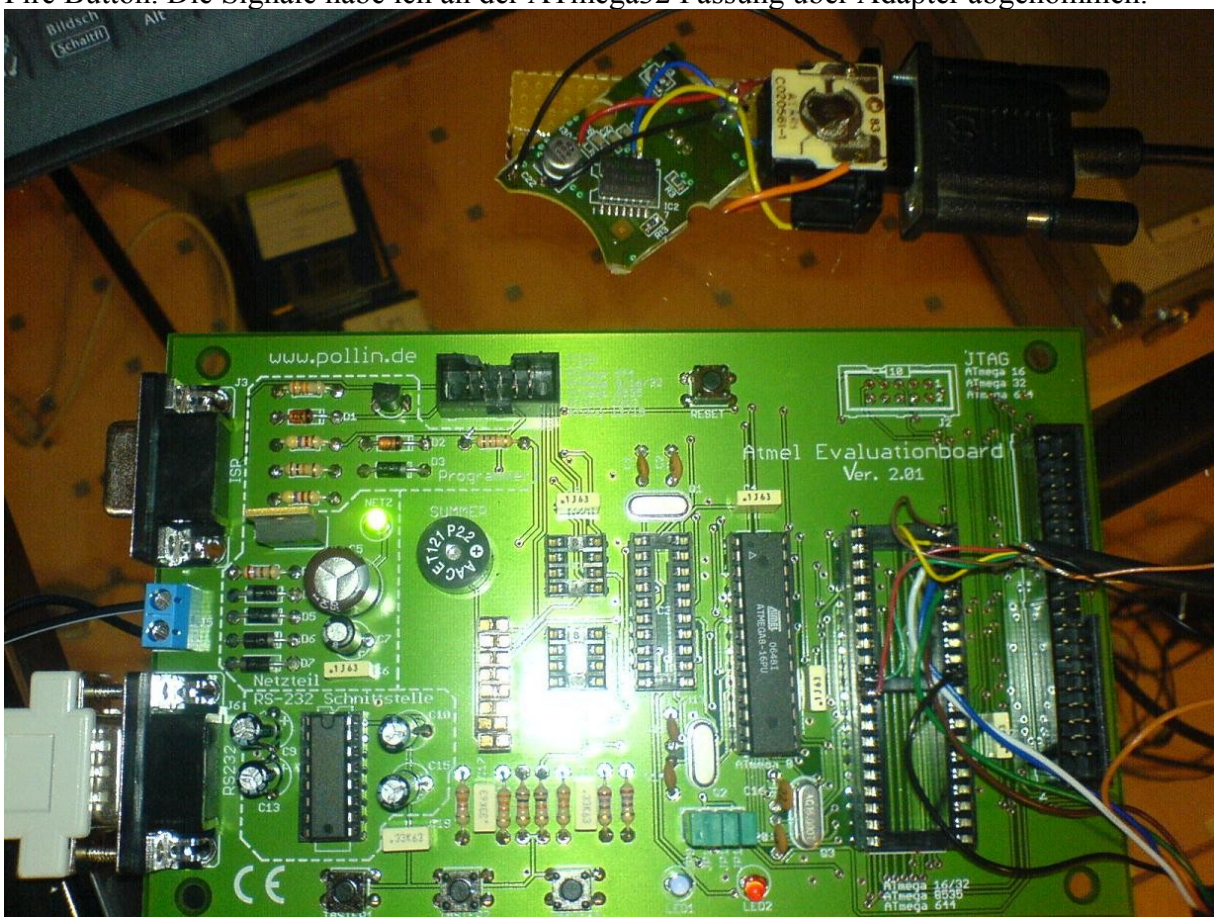


Hier von der Unterseite:



Jetzt muß dieser Sensor nur noch über einen AVR als Wandler mit einem Joystickport verbunden werden. Ab einer bestimmten Neigung wird dann der entsprechende Joystick Pin auf GND gelegt (das Gleiche wie Joystickschalter geschlossen).

Hier der Testaufbau auf einem Pollin Evaluation Board mit Atmega8. Im Bild oben ist den Sensor. Anschluß über 9-pol Sub-D (damit as auch noch am Palm PDA passt). Oben drauf der Fire Button. Die Signale habe ich an der ATmega32 Fassung über Adapter abgenommen.



Hier das Bascom Test-Programm:

```
$regfile = "m8def.dat"     'definieren des verwendeten Chips

$crystal = 1000000         'interner 1MHz Takt (default)
$baud = 1200               'definieren Baudrate der RS232 Schnittstelle

Dim X As Word
Dim Y As Word

Config Pinc.2 = Output
Config Pinc.3 = Output
Config Pinc.4 = Output
Config Pinc.5 = Output

Main:
Do
  Pulsein X , Pinc , 0 , 1
  Pulsein Y , Pinc , 1 , 1

  Print "X=" ; X ; " - Y=" ; Y
  Waitms 100

  If X > 21 Then
    Portc.5 = 0
  Else
    Portc.5 = 1
```

```
  End If
  If X < 19 Then
    Portc.4 = 0
  Else
    Portc.4 = 1
  End If

  If Y > 21 Then
    Portc.2 = 0
  Else
    Portc.2 = 1
  End If
  If Y < 19 Then
    Portc.3 = 0
  Else
    Portc.3 = 1
  End If
Loop
```

Die serielle Ausgabe diente nur zur Kontrolle. Die Auflösung in Bascom - bei internem 1MHz Takt ist ein wenig gering. In der Nullposition wird 20 zurück geliefert. Der Wert geht dann jeweils um +/- 5 hoch bzw. runter. Also 15-20-25. Evtl. kann die Waitms auch noch raus bzw. kleiner, oder man läßt den AVR mit Quartz und höherem Takt laufen. Die PIN's sollten besser Open Collector bei 1 sein! Also noch eine Menge zu optimieren.

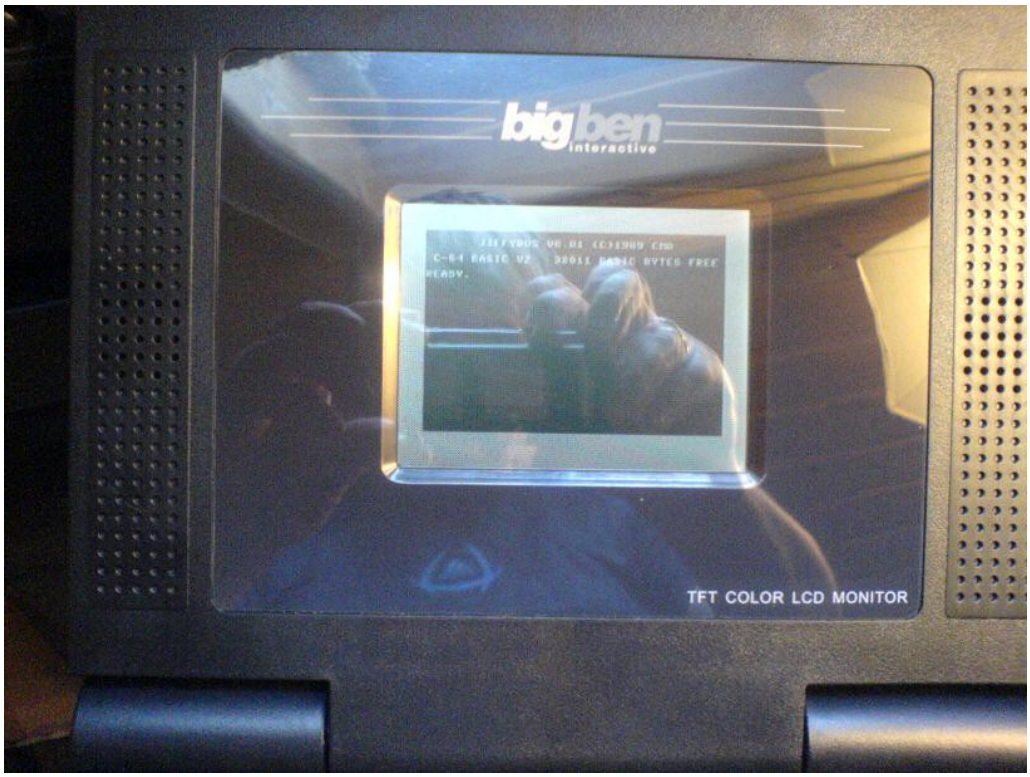
```
C.0 = X-in
C.1 = Y-in
C.2 = Up
C.3 = Down
C.4 = Left
C.5 = Right
Fire wird direkt durchgereicht
```

TFT's – Kontrollmonitore für C64

Der Betrieb eines C64 an einem TFT Fernseher ist nicht unproblematisch. Zum Kauf sollte man seinen C64 mitnehmen und im Geschäft testen ob es überhaupt geht und die Bildqualität ausreichend ist.

Für Testzwecke, bzw. als Kontrollmonitore in der Reparaturecke kann man auch die TFT-Displays der XBOX – Farbdarstellung und ohne Umbau direkt verwendbar - oder das der PS2 – Umbau nötig und nur S/W-Darstellung verwendet. Insbesondere letzteres ist 2009 für wenige Euro ‚verramscht' worden.

PS2-TFT:



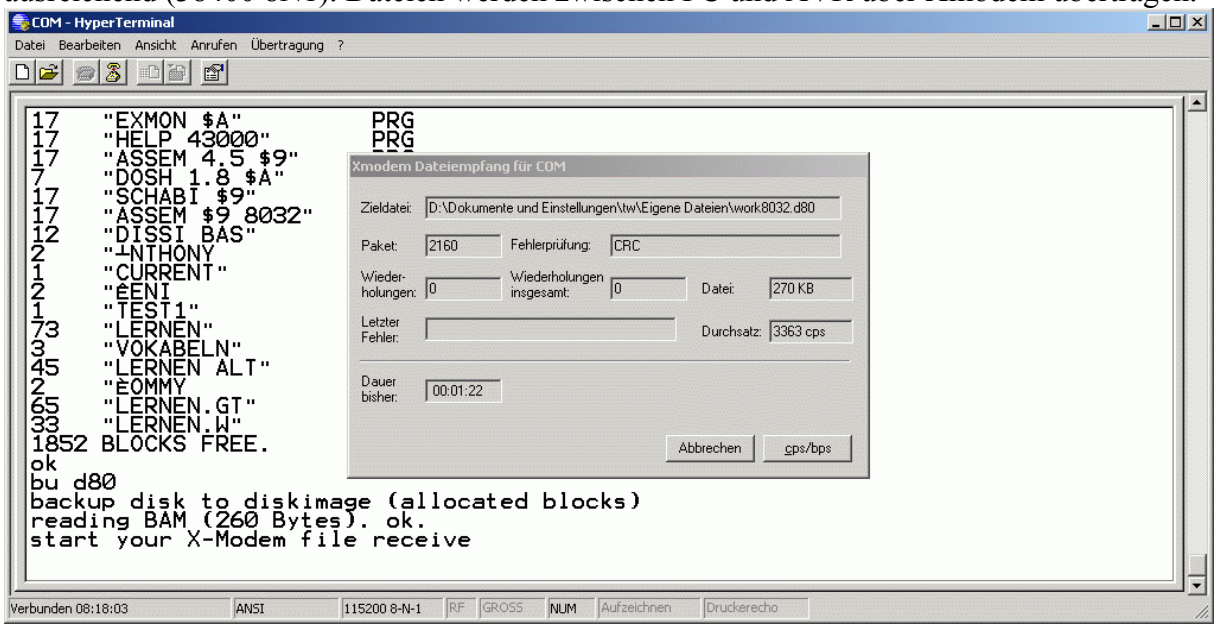
Pin	Farbe	Funktion
1	braun	Ground
2	violett	Audio rechts
3	weiss	Ground
4	grau	Audio links
5	grün	S-Video Y
6	----	
7	----	
8	blau	Video Ground
9	rot	Blau
10	---	
11	gelb	Rot
12	orange	Grün

Für BAS: Pins 5, 9,11+12 verbinden = Signal; 8=GND

XS1541 – AVR zwischen PC und CBM Floppy

Nach Ing. Thomas Winkler

Mit diesem neuen Wandler zwischen den Welten auf AVR Atmega32 Basis, lassen sich sowohl serielle IEC Floppies, als auch parallele IEEE Floppies untereinander und mit einem PC verbinden. Dabei ist auf PC Seite keine spezielle Software erforderlich. Hyperterm ist ausreichend (38400 8N1). Dateien werden zwischen PC und AVR über Xmodem übertragen.



Ich habe den XS1541 Adapter auf einem Pollin AVR Evaluation Board aufgebaut. Dort ist schon alles drauf, was benötigt wird:

- Atmega32 mit Programmiermöglichkeit.
- MAX232 als RS232 Pegelwandler an RX+TX
- 16MHz Quartzschaltung.

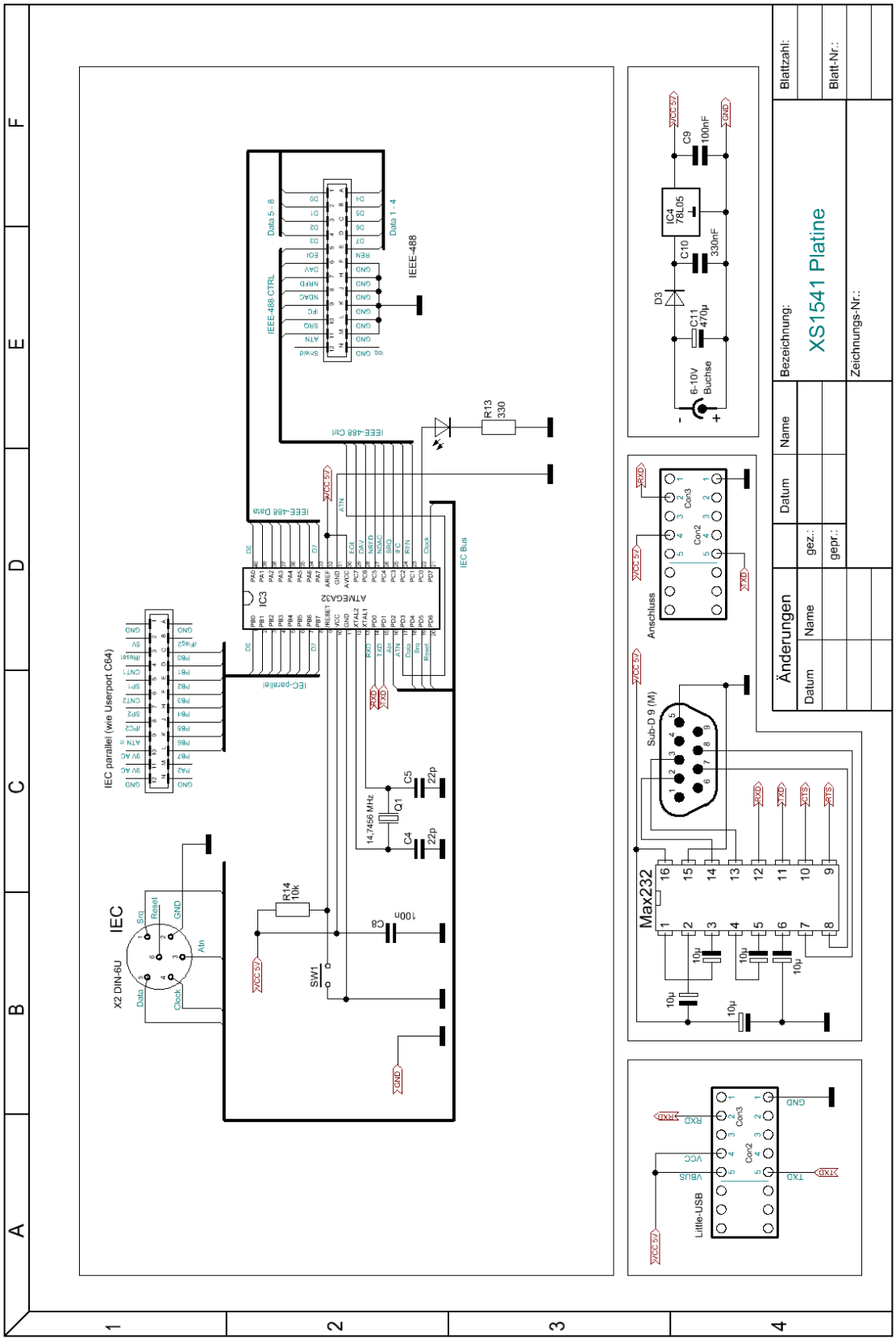
Zum seriellen IEC Anschluß (siehe mmc2iec Kapitel) müssen anschließend folgende Verbindungen hergestellt werden:

AVR	PIN	Signal	J4 Anschlußpunkt auf Pollin Board
pd7	21	clock	33
pd6	20	reset (nicht verbunden)	32
pd5	19	srq	31
pd4	18	data	30
pd2	16	atn	28
+gnd		Masse	35,37,39

Bild des XS1541 realisiert auf dem Pollin Board:

Firmware unter: http://petersieg.webng.com/xs1541/xs1541.zip

Schaltplan:



Befehls Syntax

Die Befehle können direkt im Terminal Programm eingegeben werden und werden direkt vom Controller ausgeführt. Dadurch ist man komplett unabhängig von der Hardware. Das funktioniert auf jedem PC, unter Windows, Linux oder Apple OS, es muß nicht mal ein PC sein.
Einstellungen: 38400 Baud 8N1 – Kein Hardware Flowcontrol – Kein Xon/Xoff
Einige Befehle kontrollieren direkt das Protokoll oder sogar die Portleitungen. Diese "low level" Befehle sind zum Test des Adapter bzw. für PC Programme gedacht.
Jeder Befehl hat einen **Mussteil** und einen **Kannteil**. Der Kannteil kann ganz oder teilweise weggelassen werden. Der Kannteil steht in rechteckigen Klammern. Manche Befehle haben mehrere Kürzel.

h[elp]

Listet alle Befehle die in der Firmware des Controller implementiert sind. Es kann 'h', 'he', 'hel' oder 'help' eingegeben werden.

iee[e]

Der aktive Bus wird zum parallelen IEEE-488 Bus geschalten. Alle Nachfolgenden Befehle (außer IEC) beziehen sich nun auf den IEEE-488 Bus. Dieser Modus ist die Standard Einstellung nach einem Controller Reset.

iec

Der aktive Bus wird zum seriellen IEC Bus geschalten. Alle nachfolgenden Befehle (außer IEEE) beziehen sich nun auf den IEC Bus.

dev[ice] oder #

Damit wird die aktive Gerätenummer des aktiven Bus festgelegt oder abgefragt. Wenn dem Befehl ein numerischer Parameter folgt wird die Gerätenummer gesetzt. Der Befehl alleine ohne Parameter listet die aktuelle Einstellung. Für beide Bus Systeme ist die voreingestellte Adresse nach einem Reset 8.

@

Dieser Befehl sendet einen Befehl an die aktive Floppy (Befehlskanal 15) oder fragt den Fehlerkanal / Status ab wenn der Befehl alleine (ohne Parameter) gesendet wird.

dir[ectory] oder **ca[talog]** oder **$**

Listet das Verzeichnis der eingelegten Diskette. Als Parameter kann optional ein String mitgegeben werden. Das Format entspricht dem des Floppy DOS (load "$....",8). Bei Doppelfloppy sollte man als Parameter zumindest das Laufwerk (0 oder 1) mitgeben, sonst wird das Verzeichnis beider Laufwerke ausgegeben, was zu einem Fehler führt wenn nicht beide Laufwerke formatierte Disketten eingelegt haben.

filec[opy] oder **fc**

Kopiert Dateien von Floppy zu Floppy. IEC zu IEEE oder umgekehrt ist möglich.

filed[ump] oder **fd**
Erstellt einen Hexdump von einer Datei.

bac[kup]

Erstellt eine Image Datei einer ganzen Diskette (d64, d72, d80, d81, d82). Die Image Datei wird per X-Modem Protokoll direkt auf die Festplatte des PC gespeichert.

res[tore]

Schreibt eine Image Datei auf eine Diskette zurück (d64, d72, d80, d81, d82). Die Image Datei wird per X-Modem Protokoll vom PC zum Adapter übertragen.

lo[ad]

Ladet eine Datei zum PC (P00, T64). Die Datei wird per X-Modem Protokoll direkt auf die Festplatte des PC gespeichert.

sa[ve]
Schreibt eine Datei auf eine Diskette zurück (P00, T64). Die Datei wird per X-Modem Protokoll vom PC zum Adapter übertragen.

Link zur Homepage des Projektes: http://xs1541.t-winkler.net/

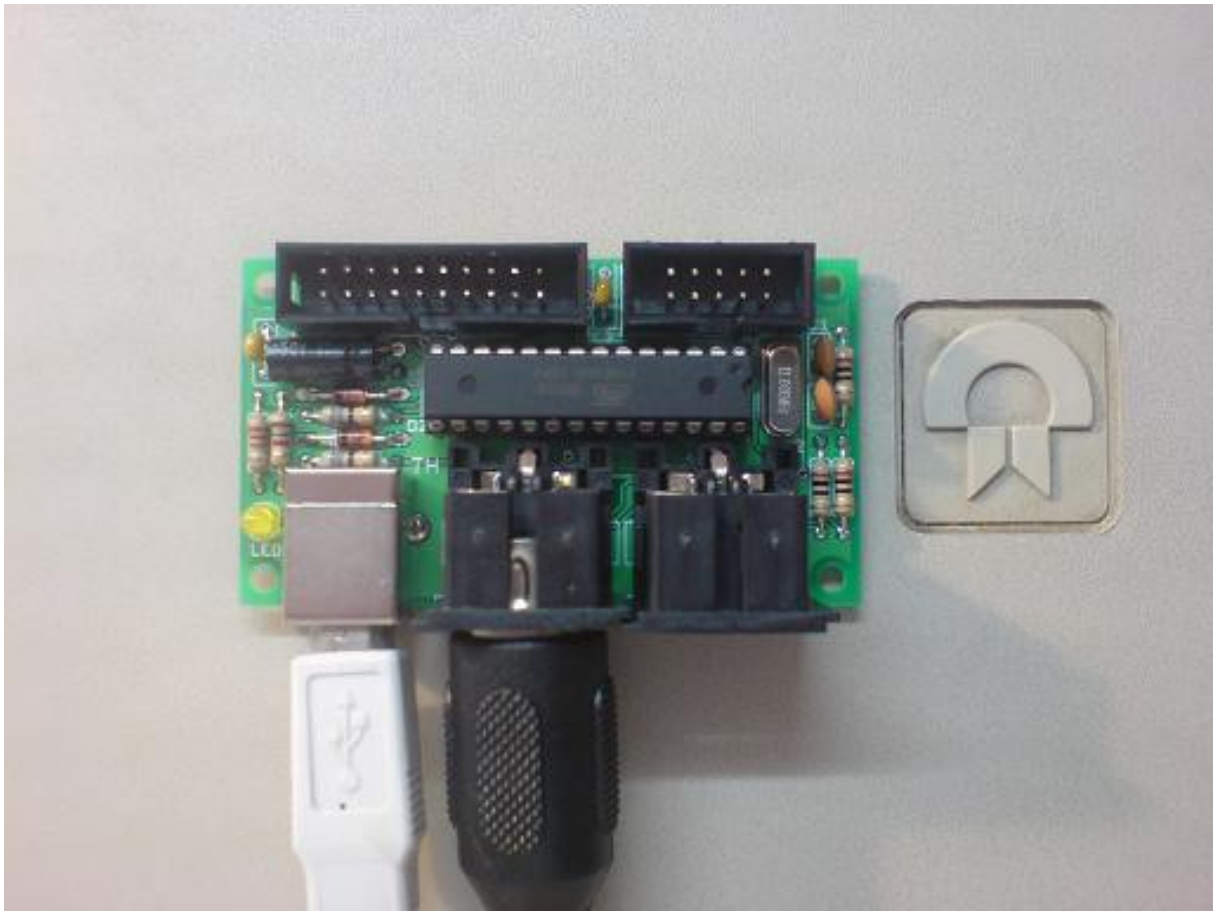
XU1541 – USB Interface für CBM Floppies

Link: http://www.trikaliotis.net/xu1541
http//opencbm.sf.net/

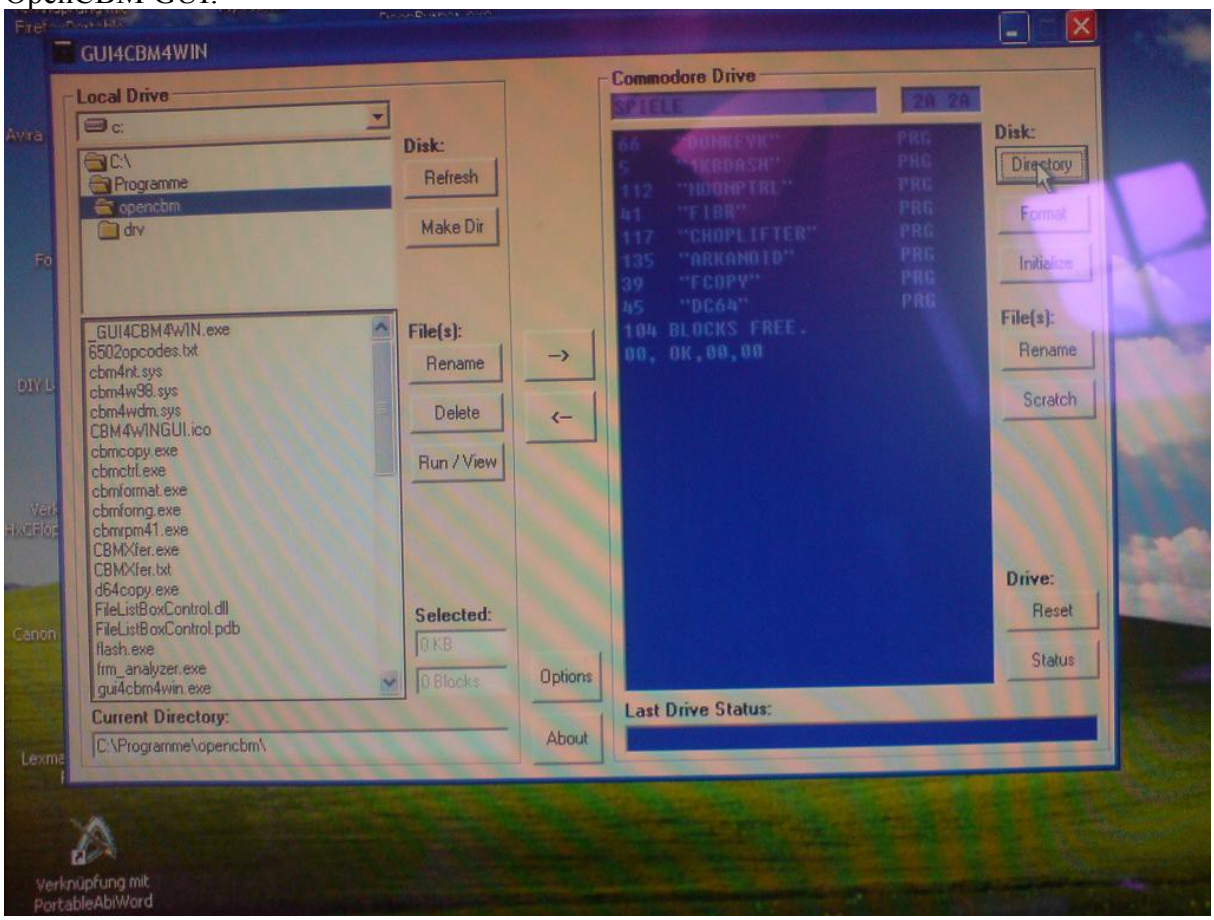
Das XU1541 Projekt wurde ursprünglich von Till Harbaum gestartet. Ende 2007 übernahm Spiro Trikaliotis die weitere Betreuung.

Da inzwischen immer weniger PC's und insbesondere Notebooks noch Parallelports haben, ersetzt XU1541 die davor eingesetzten X*1541 Kabel und ermöglicht so den Anschluß von IEC Floppies an den USB Bus. Dabei wird eine CBM Floppy allerdings nicht zum universiellen Wechseldatenträger, sondern wird nur über spezielle Low-Level Befehle mit spezieller Software über USB ansprechbar. Die Softwareunterstützung für XU1541 bildet das OpenCBM Projekt.

XU1541 Interface:



OpenCBM GUI:



Das OpenCBM Projekt stellt dabei einige Kommandozeilentools zur Verfügung.

z.B d64copy.exe zum transferieren von D64 Images von und zur Floppy:

Image schreiben: d64copy <image.d64> 8
Image lesen: d64copy 8 <image.d64>

XUM1541 – Zoomfloppy/Teensy/Open Kubus

Links:
http://www.root.org/~nate/c64/xum1541/
http://xs1541.t-winkler.net/zip/Zoomfloppy/

Das XUM1541 Interface ist der Nachfolger des XU1541 USB CBM Interfaces.

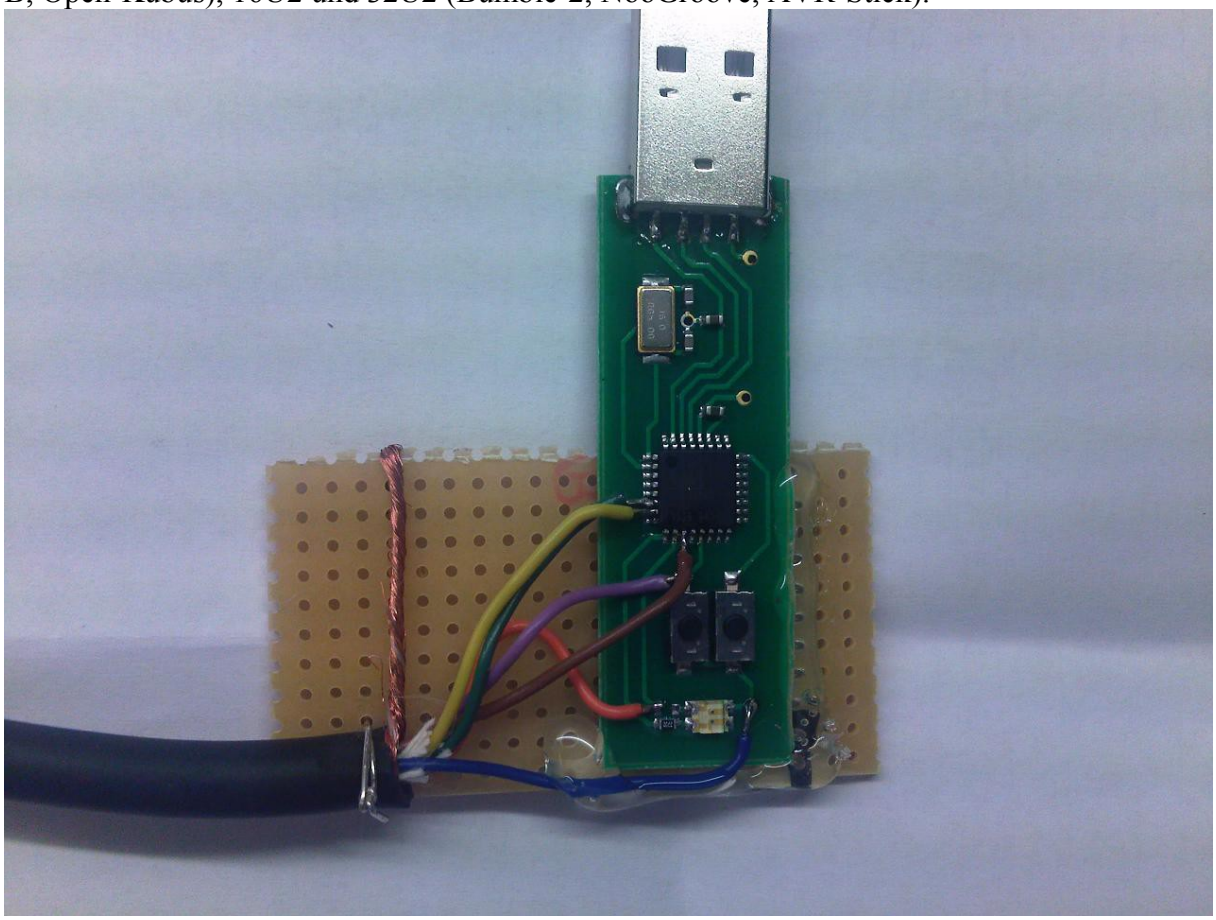
Nate Lawson, der großartige Erfinder des XUM-1541 (Zoomfloppy) hat den Support für die Billigversion des Zoomfloppy in Form von AT90USB162 Boards bereitgestellt. Jeder der ein bisschen basteln (löten) kann, kann sich nun selbst ein Zoomfloppy auf Teensy Basis selbst bauen.

Das Teensy oder eines seiner zig Clones (Bumble-B, Open Kubus ...) kann man sich sehr günstig (<20€) im Internet bestellen.

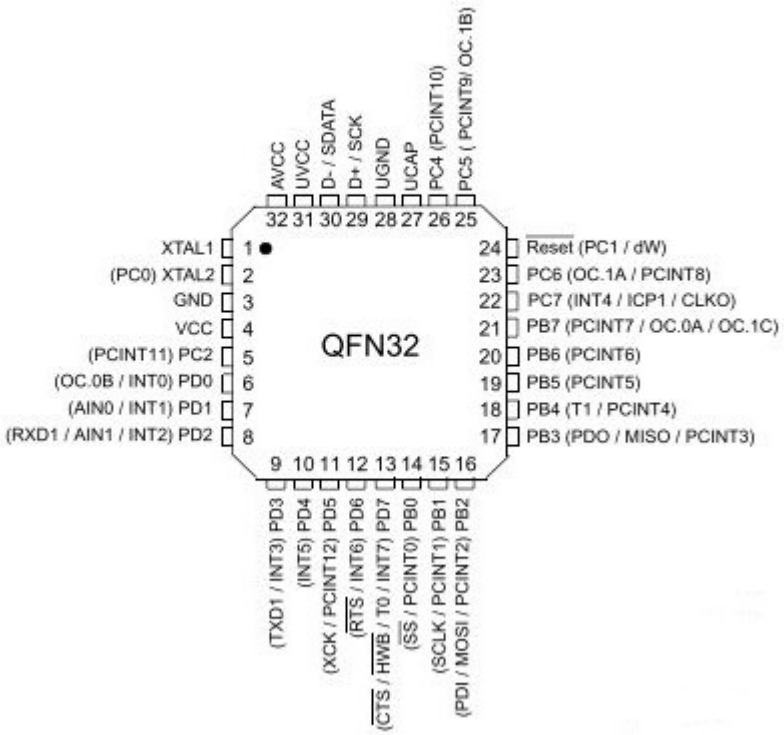
Dazu benötigt man noch ein halbes IEC Kabel, eine Zange und ein Messer zum abisolieren. Das Kabelende lötet man auf das Teensy (siehe Bild) und befestigt es noch mit einem Kabelbinder oder anderem Zugschutz. Fertig ist das neue USB CBM Floppy Interface.

Die Idee und die Durchführung zu der Teensy Idee stammt von Thomas Kindler.

Supported sind zur Zeit Sticks und Boards mit den AVR Controllern AT90USB162 (Bumble-B, Open-Kubus), 16U2 und 32U2 (Bumble-2, NooGroove, AVR-Stick).



AT90USB162 QFN32 Pinout:

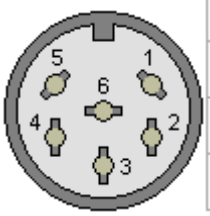


Pin-Belegung für XUM1541:

+ PD0 DATA
+ PD1 CLK
+ PD4 ATN
+ PD5 SRQ
+ PD7 RESET
+
+ PB0..7 DATA 0..7 Optionale parallel Verbindung
+
+ PD6 LED

Ebenso muss Masse/GND mit dem IEC Gerät verbunden werden!

IEC Pinbelegung:



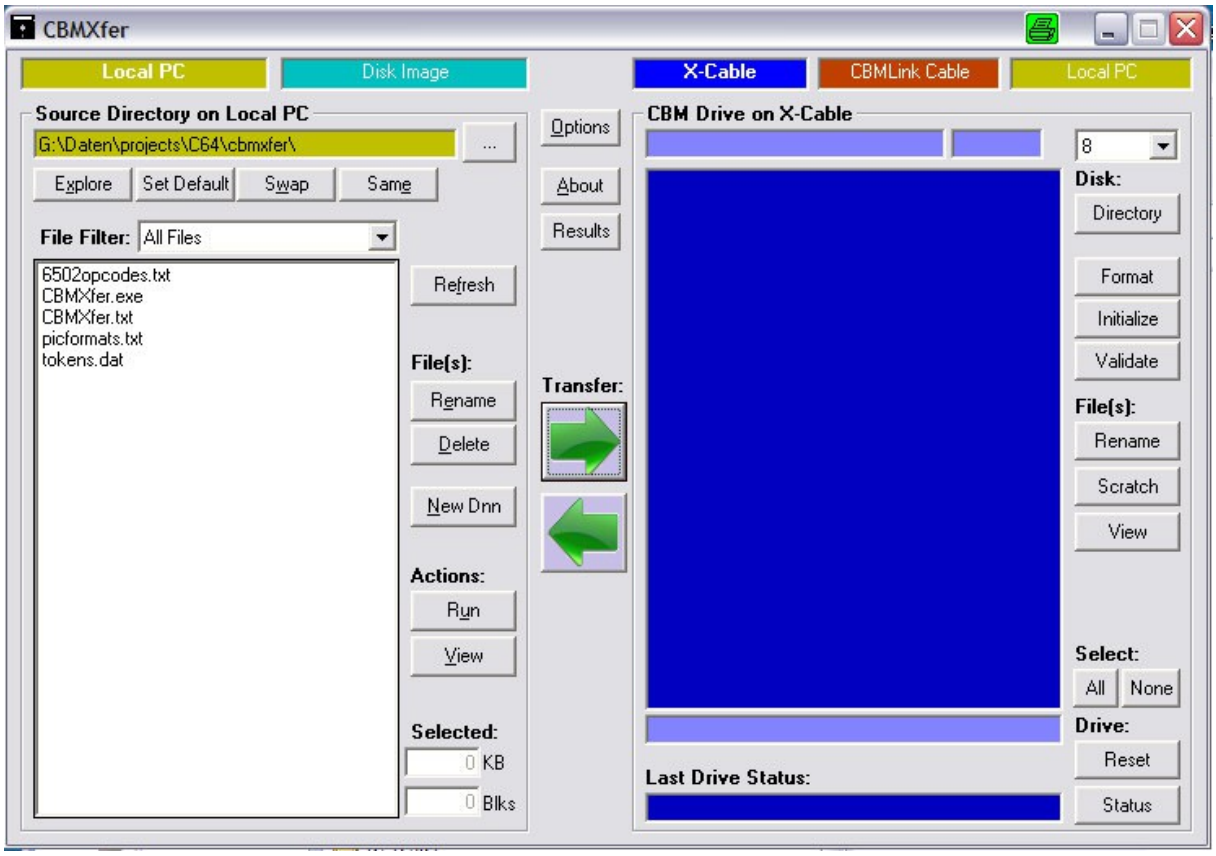
Pin	Signal
1	SERIAL /SRQIN
2	GND
3	SERIAL ATN IN/OUT
4	SERIAL CLK IN/OUT
5	SERIAL DATA IN/OUT
6	/RESET

Das richtige HEX File für alle AT90USB162 Boards (zb. Bumble) ist das CHEAP1.
Das richtige HEX File für alle Atmega32U2 Boards (zb. NooGroove) ist das CHEAP2.

Dieses HEX File ist über das Atmel Tool Flip in den AVR zu flashen. Siehe dazu die Anleitung von Flip.

Auch hier ist das Frontend zur Bedienung dann OpenCBM (siehe XU1541) mit seinen Kommandozeilen Programmen (cbmctrl; d64copy; etc), oder alternativ ein Frontend wie z.B CBMXFER.

Link: http://www.6502.org/users/sjgray/software/cbmxfer/cbmxfer.html

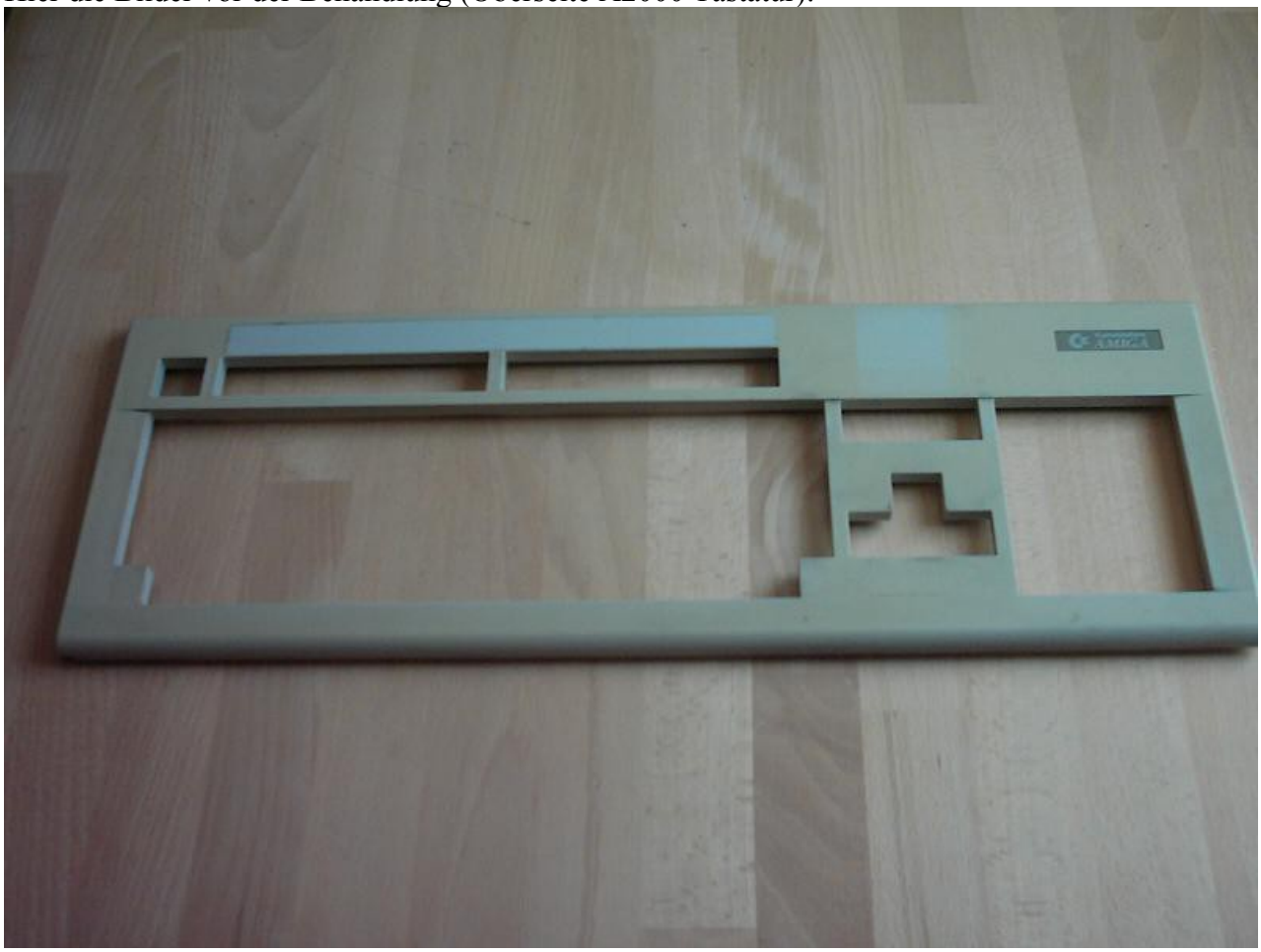


Kampf dem Gilb

Oder wie man vergilbte Plastikteile wieder hell bekommt.

Wer kennt sie nicht, die häßlich vergilbten Plastikgehäuse unserer Schätzchen..
Aus dem forum64, dem a1k.org forum hatte ich einige Angaben und Tipps gelesen..
Ausprobiert habe ich es dann mit Wasserstoffperoxid 9%. Es geht auch 12%. Ich habe dabei ein Produkt aus dem Friseurbedarf verwendet, welches eine cremige Konsistenz hatte.

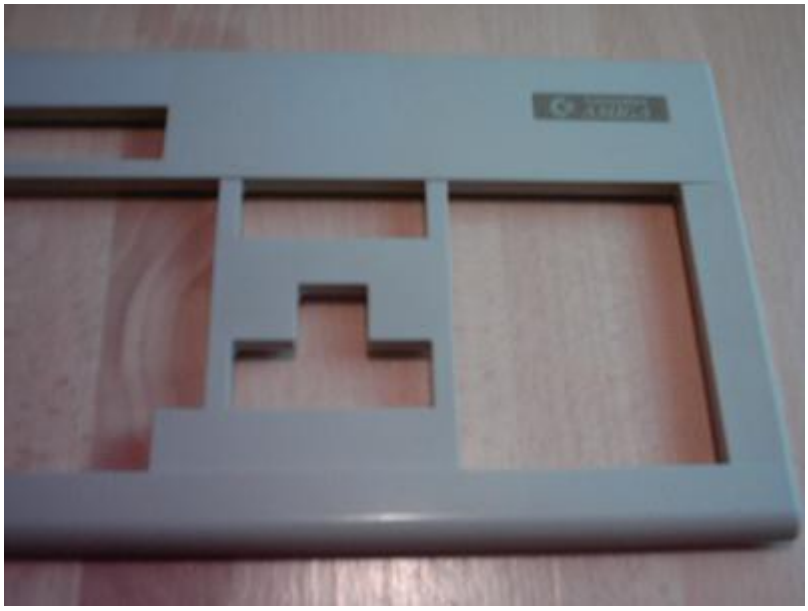
Hier die Bilder vor der Behandlung (Oberseite A2000 Tastatur):



Die Behandlung:

Zuerst gut reinigen mit Wasser und Seife. Trocknen. Danach auf einer Unterlage, draußen an der Sonne, mit dem Mittel satt einpinseln. 3-4 Stunden unter Sonnenstrahlung einwirken lassen. Unter viel Wasser abspülen und trocknen.. Je nach Bleichungsgrad ggf. die Behandlung wiederholen..

Hier die Bilder nach der Behandlung:



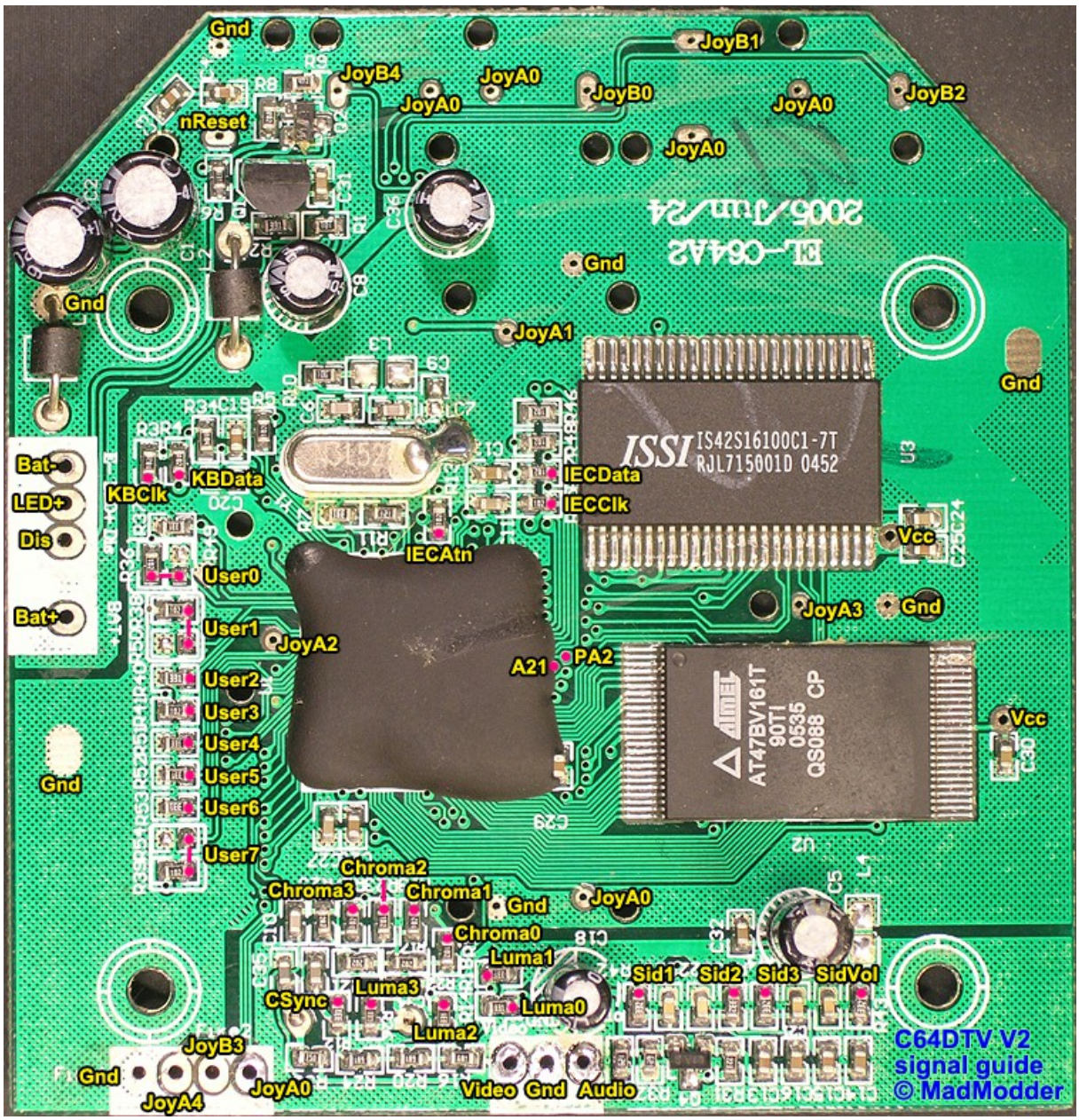
64 DTV

Das 64 DTV ist die Consumer Inkarnation der erfolgreichen Bemühungen einen C64 in FPGA/ASIC zu implementieren. Die FPGA Entwicklungsstufe war der C-One. Mit nur 3 Chips, ASIC (mit CPU, CIA's, SID, VIC), 2MB Flash Rom (mit Kernal, Basic, Char und Programmen) und Ram, konnte ein kompletter C64 inkl. Programmen in einen nachempfundenen Competition Pro Joystick nachgebaut werden. Die NTSC-Variante, welche sich vom Softwareumfang und auch in der Hardware unterscheidet wird hier nicht betrachtet. In der PAL-Variante ist z.B. Alleykat hinzu gekommen, während World Games weggefallen ist.

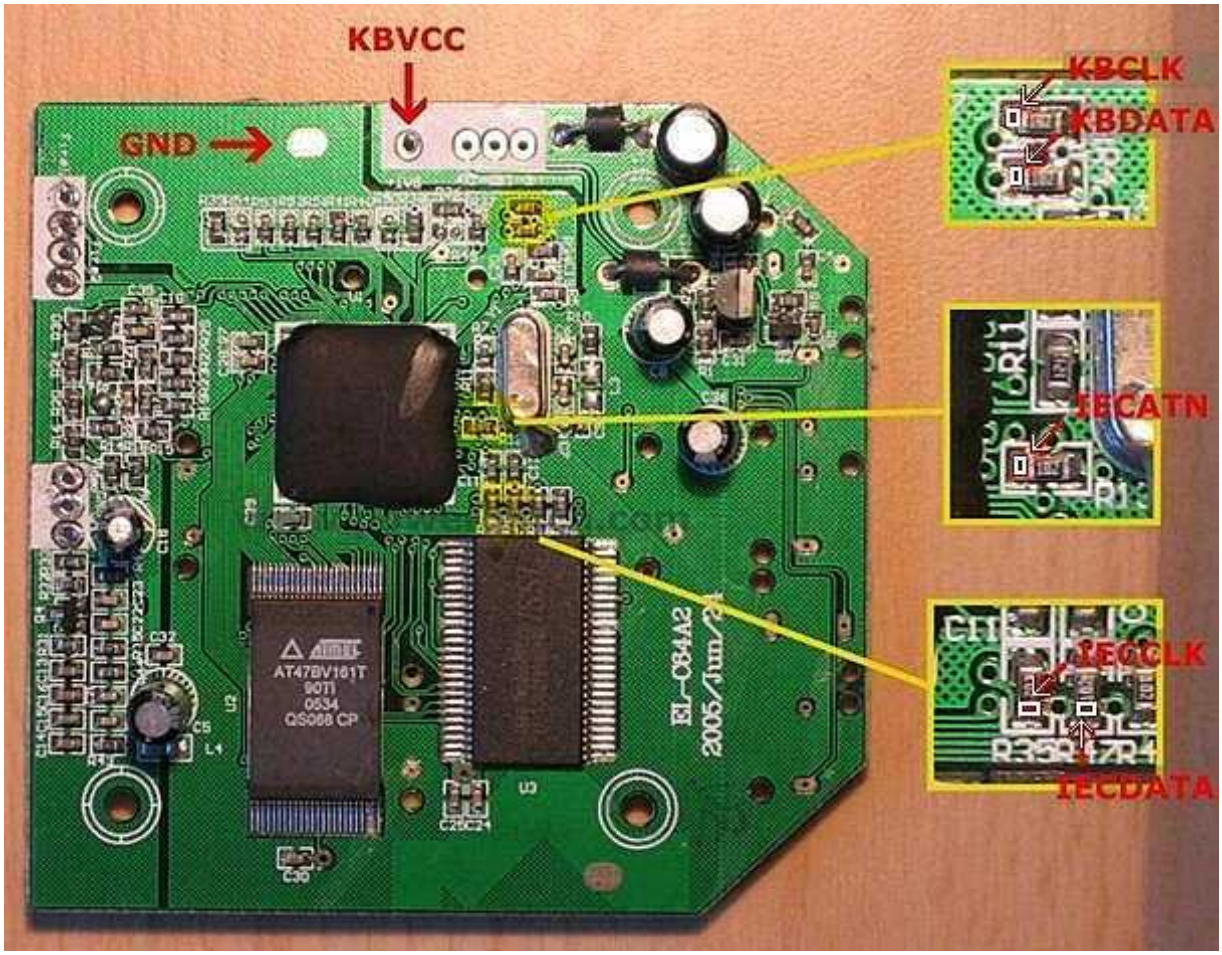
Das 64DTV (DTV=Direct To TV) - so haben findige Hacker schon bald herausgefunden - kann um so einiges erweitert und modifiziert werden. Man kann z.B eine PS/2 Tastatur anschließen. Durch halten von 'STRG' beim Einschalten kommt man so direkt ins Basic! Man kann das DTV um einen IEC Anschluß erweitern und so z.B orig. Floppies oder MMC2IEC anschließen. Weiterhin können Joystickport Anschlüße angebracht anwerden, um bessere und austauschbare Joysticks nutzen zu können. Darüber hinaus kann auch das 2MB Flash verändert werden.. Zum Beispiel um weitere oder andere Spiele dauerhaft im Flash zu haben. Für weitere Informationen helfen sicher Suchmaschinen.. oder die Links in diesem Kapitel.

DTV Anschlußpunkte im Überblick

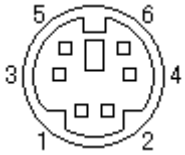
(von Daniel Svegert / MadModder):



Anschlußpunkte IEC + PS/2

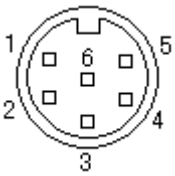


PS/2-Keyboard-Stecker: (Ansicht auf Steckerfront(m) (bzw. Lötseite Kupplung(f)))



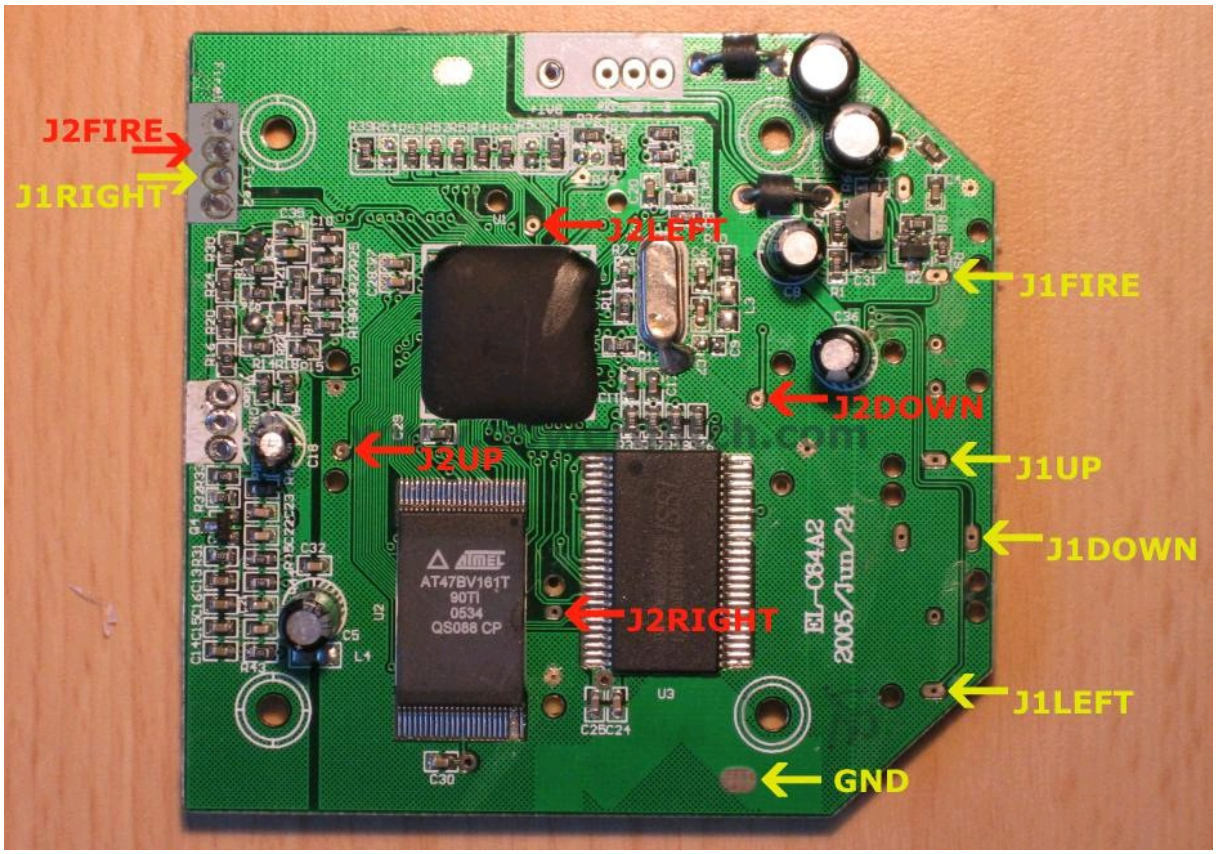
1. DATA
2. N.C.
3. GND
4. +5V
5. CLOCK
6. N.C.

IEC-(Floppy-)Stecker: (Ansicht auf Steckerfront(m) (bzw. Lötseite Kupplung(f)))

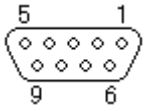


- 1. SRQ (unbeschaltet)
  2. GND
  3. ATN
  4. CLOCK
  5. DATA
  6. RESET (unbeschaltet)

Anschlußpunkte Joysticks

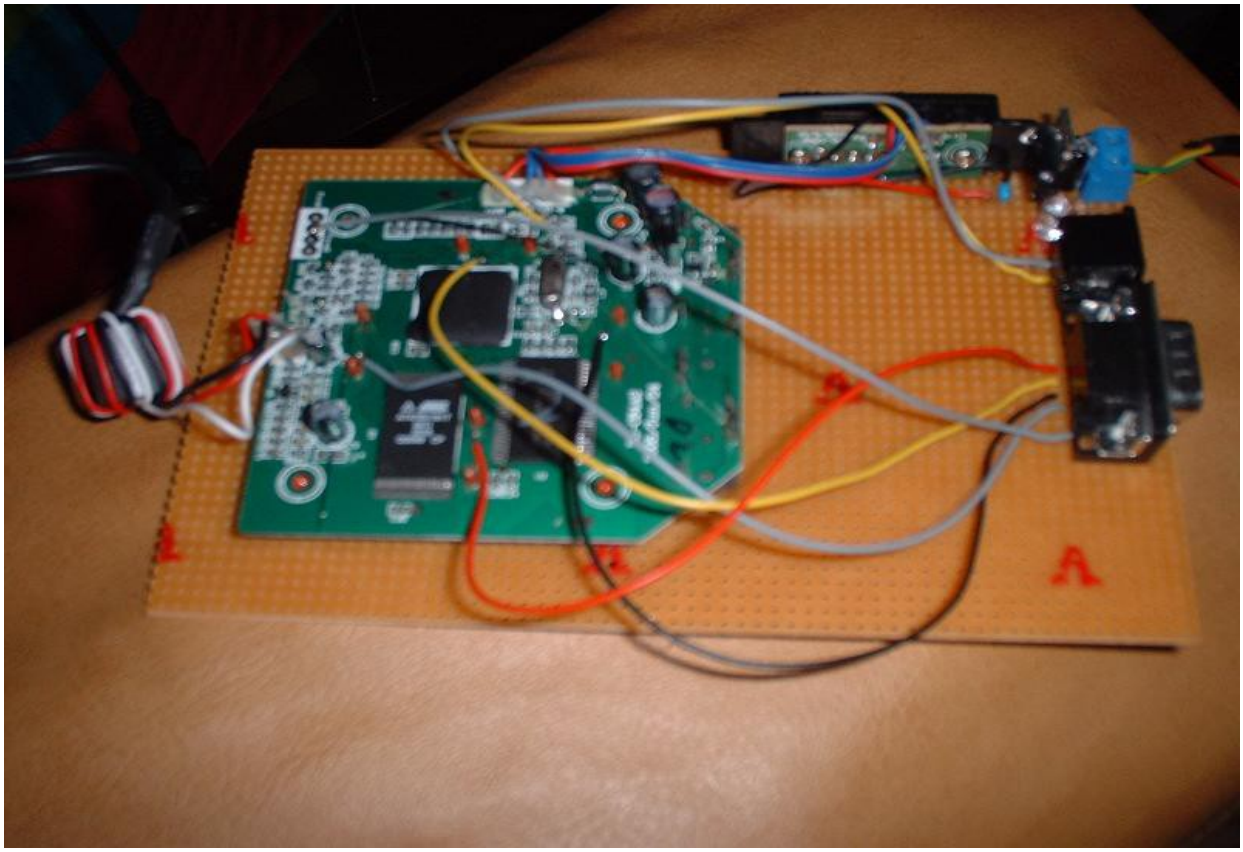


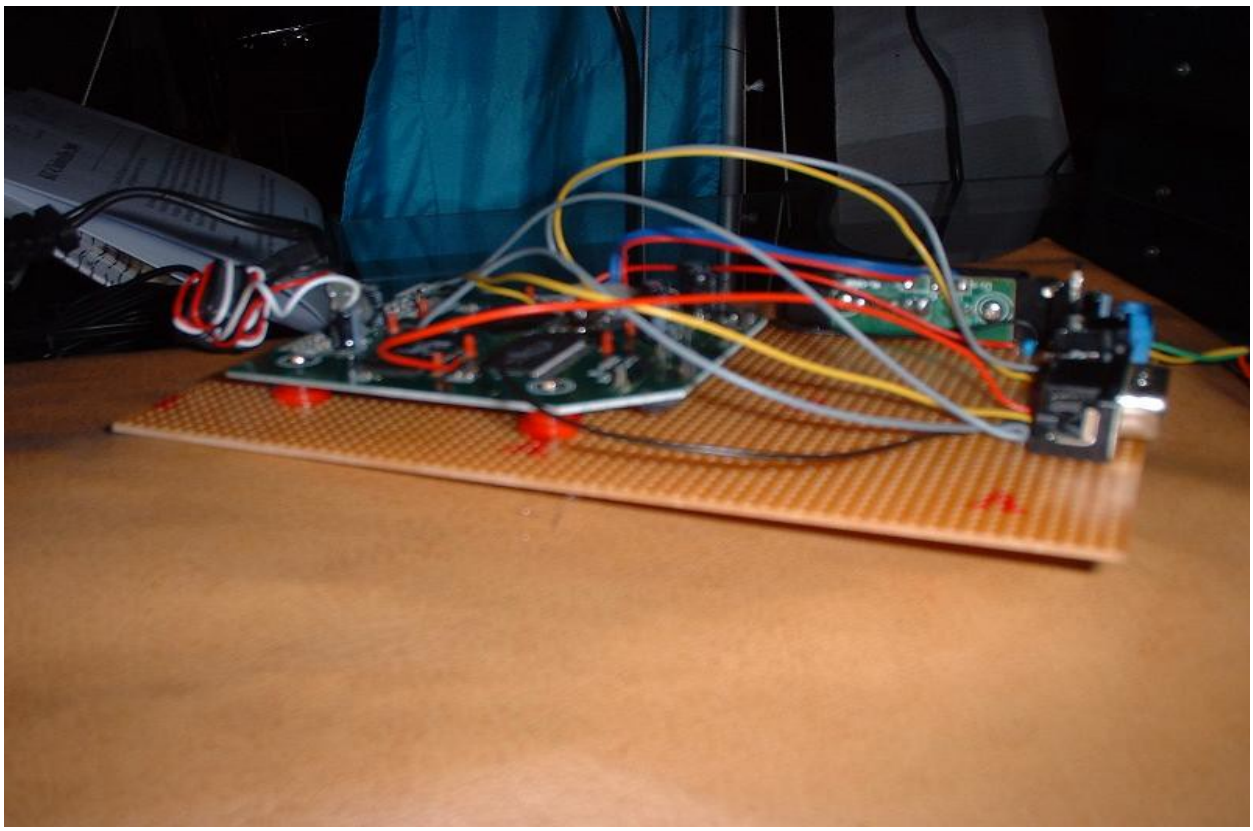
Joystick-Stecker: (Ansicht auf Kupplungsfront(f) (bzw. Lötseite Einbaustecker(m)))



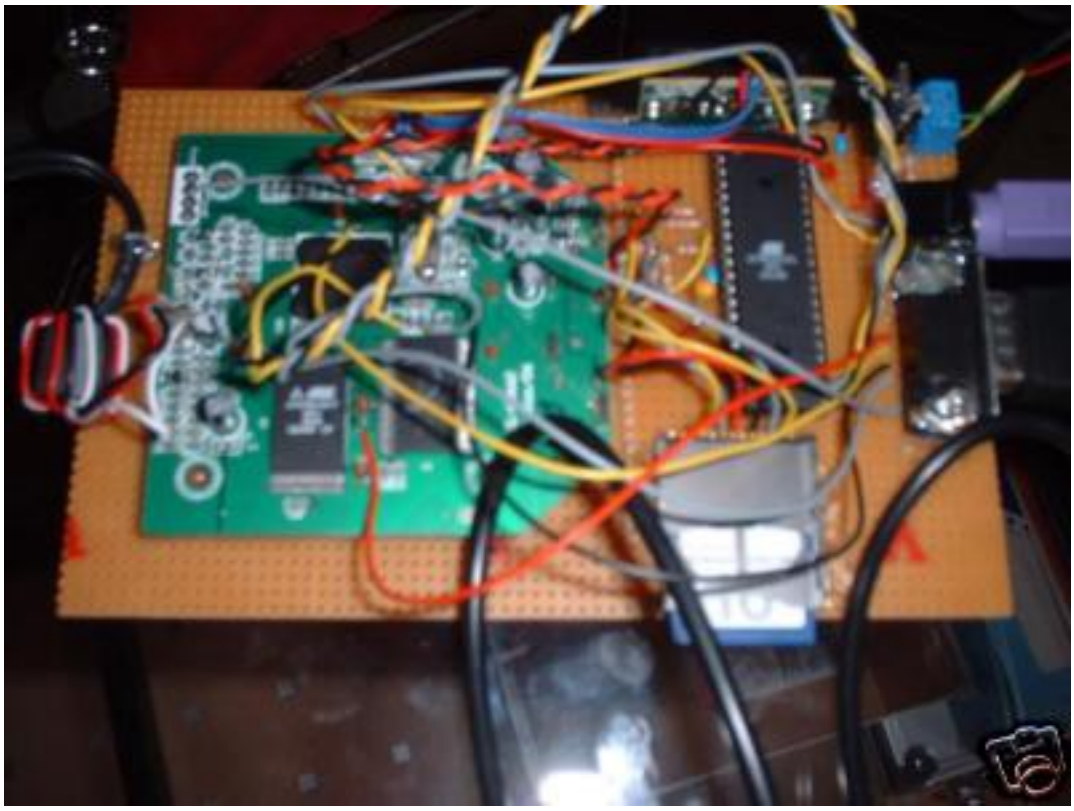
1. UP
2. DOWN
3. RIGHT
4. LEFT
5. (POTY)
6. FIRE
7. +5V
8. GND
9. (POTX)

Hier Bilder meines ersten Mods. Lochrasterplatine 160x100mm als Grundträger. Rechts: 5V, PS/2, Joy2:





Hier inklusive dem MMC2IEC Gerät:



Easyflash

Dieses wunderbare Tool von Roland Tögel, ermöglicht es auf sehr einfache Weise sein DTV mit einigen neuen Spielen zu bestücken. Dabei bleiben alle originel Spiele, Dateien und das original Menü weiterhin zugänglich. Es werden max. 768 Blocks im hinteren, freien Bereich mit Spielen belegt und ein neues Menu installiert. Aus diesem Menu sind direkt die neuen Spiele und das original Menu aufrufbar. Außerdem sind durch das neue Menu, die zeitintensiven Intro's ebenfalls entfallen, sodaß der Start erheblich beschleunigt ist. Ich habe mit diesem Tool selbst gearbeitet und alles hat wunderbar funktioniert. Es ist allerdings nur für die PAL Version mit AT47... Flashchip nutzbar.

Dies Tool ist ideal, wenn man schon eine Floppy oder MMC2IEC am DTV angeschlossen hat. Dann das Tool und die gewünschten Programme in PRG oder codierter DTV Form auf Floppy oder SD Karte aufspielen, Tool starten und ‚L' zum laden von Standard C64 Programmen oder ‚E' für schon codierte DTV Programme. Zuerst den Dateinamen, so wie er auf dem Datenträger ist eingeben. Nach dem Laden den Namen, den das Programm/Spiel im neuen Menü haben soll (wird mit dem Dateinamen vorbelegt.. man kann also auch einfach Return drücken), danach die Startadresse, oder einfach 0100 eingeben. Die vierte Eingabe kann leer bleiben. Das für alle Programme wiederholen, bis max. der verfügbare Platz von 768 Blocks erschöpft ist. Dann erst ‚F' für das flashen der neuen Programme und des entsprechenden Menüs engeben. Beim flashen sollte man immer Sicherstellen, das es keine Unterbrechung der Stromzufuhr geben kann! Ebenfalls sollte man am besten nur sogenannte One-filer Programme verwenden. Das sind Programme die selbst alles beinhalten, was zum Ablauf benötigt wird. Also keine separaten Dateien außer der PRG Datei selbst brauchen.

Das Tool gibt es in der CSDB:
http://noname.c64.org/csdb/release/?id=41413&show=review#review

Links

http://picobay.com/dtv_wiki/index.php?title=C64_DTV_Hacking_Wiki
http://www.powerglitch.com/index_dtv.html
http://www.geocities.com/dtvhacking/
http://www.kahlin.net/daniel/dtv/
http://galaxy22.dyndns.org/dtv/
http://www.geocities.com/homeofoscarvermeulen/dtv/dtv_pal.html

Amiga 500

Der Amiga 500 ist der meistverkaufte Amiga-Computer von Commodore. Er wurde 1987 auf der CeBit mit dem Amiga 2000 vorgestellt und besaß ähnliche Leistungsdaten wie dieser. Zielgruppe für den Rechner waren vor allem Privatanwender mit kleinerem Geldbeutel. Zu Lasten der Aufrüstbarkeit wurde auf ein Desktop- bzw. Towergehäuse verzichtet und stattdessen ein Design gewählt, das dem des Commodore 128 ähnelte. Der Amiga 500 war – mehr noch als der Amiga 2000 – zu seiner Zeit besonders als Spielecomputer beliebt, da Bild- und Tonqualität dem damals üblichen PC deutlich überlegen waren.

Der Amiga 500 besaß eine mit 7,09 MHz (PAL-Variante) bzw. 7,14 MHz (NTSC-Variante) getaktete Motorola 68000-CPU und 512 KByte Arbeitsspeicher. Dieser konnte mittels eines Erweiterungs-Slots bis auf 2,8 MByte erweitert werden. In den Jahren 1987 und 1988 wurde der Amiga 500 mit dem Amiga-Betriebssystem AmigaOS 1.2 (Kickstart) ausgeliefert, ab 1988 wurde dieses durch die wesentlich stabilere Version 1.3 abgelöst.

Wie beim Amiga 1200 und dem C64 muss an den Amiga 500 ein spezielles Netzteil angeschlossen werden – im Amiga 500 ist kein Netzschalter eingebaut.

Amiga 600

Der Amiga 600 war der erste Commodore Rechner, der SMD Technik nutzte und konnte daher viel kleiner als ein A500 gebaut werden.

Als Besonderheit, besaß er einen PCMCIA Slot. Dank einer aktuelleren Kickstartversion und mittels PCMCIA – CF Kartenadapter lassen sich Amiga Diskettenimages vom PC auf Amiga Floppies zurückschreiben.

Amiga Custom Chips

Hier als kleine Hilfestellung die häufigstens Amiga Spezialchips im Überblick.

Agnus - Für Bildschirmaufbau (mit Denise)

8370 (NTSC)	CPN: 318070-01
8361 (NTSC)	CPN: 252125-01
8371 (PAL)	CPN: 318071-01
8367R0 (PAL)	
8372A (Fat Agnus)	CPN: 318069-01
8375 (Fat Agnus)	CPN: 390544-01
	CPN: 318069-10 (A600)
8372B (Super Agnus)	CPN: 318069-03
	2048KB of memory

CIA - Complex Interface Adapter - Parallel, Seriell, Tastatur, Floppy

8520	CPN: 391078-01/03

Denise - Mit Agnus für Bildschirmaufbau

8362R5 (Denise)	
8362R6 (Denise)	CPN: 252126-01
8362R8 (Denise)	CPN: 252126-02
8373R4 (Super Denise)	CPN: 390433-01
CPN: 391081-10 (A600)	

Gary - I/O Prozessor - Bus Access, Floppy, IDE

5719R2 (Gary)	CPN: 318072-01
41-Fat Gary (Fat Gary)	CPN: 390540-02

Paula - Sound, UART, Interupt

8364	CPN: 252127-01
8364R7	CPN: 252127-02
	CPN: 391077-01

Im A3000/4000/1200 gab es weitere Spezialchips:

Alice – 8374	CPN: 391010-01
Amber	CPN: 390538-03
Budgie	CPN: 391425-01
Buster	CPN: 390539-11
DMAC	CPN: 390537-02
Gayle	CPN: 391424-02
Lisa	CPN: 391227-01
Ramsey	CPN: 390541-04

CPN = Commodore Part Number)

Amiga Fehlercodes nach dem Einschalten

Beim Starten führt der Amiga verschiedene Selbsttests durch und zeigt den Status über die Bildschirmfarbe und die Tastatur-LED's an.

Ändert sich der Bildschirm von Dunkelgrau über Hellgrau nach Weiß, ist alles in Ordnung. Ohne bootfähige Diskette bzw. Festplatte am System, sollte kurz danach ein Diskettensymbol Am Bildschirm dazu auffordern eine entsprechende Diskette einzulegen.

Ansonsten können folgende Fehleranzeigen ggf. das Problem eingrenzen helfen. Generell muß ein entsprechender Baustein nicht gleich defekt sein! Oft sind es auch einfach nur Kontaktprobleme. Chip – wenn gesockelt – entfernen, Pin's säubern und wieder einsetzen..

Farbcode	Diagnose
Dunkel Grau	Prozessor OK.
Hell Grau	ROM-Prüfsumme OK.
Weiß	Arbeitsspeicher (RAM) reicht aus (mindestens 256 KByte Chip-RAM und falls vorhanden, mindestens 256 KByte Fast-RAM) - das System startet…
Rot	Fehler Kickstart-ROM. Kontakte prüfen. Sitzt das Rom gut im Sockel?
Grün	Fehler in den unteren 256k Chip-RAM. Das können die Rambausteine selbst, der Agnus (8371/2/5) oder sehr selten auch der Gary Chip (5719) sein.
Blau	Fehler bei den Custom-Chips.
Gelb	Ein Softwarefehler ist aufgetreten, bevor die Fehlerbehandlungsroutinen („Guru Meditation") installiert wurden.

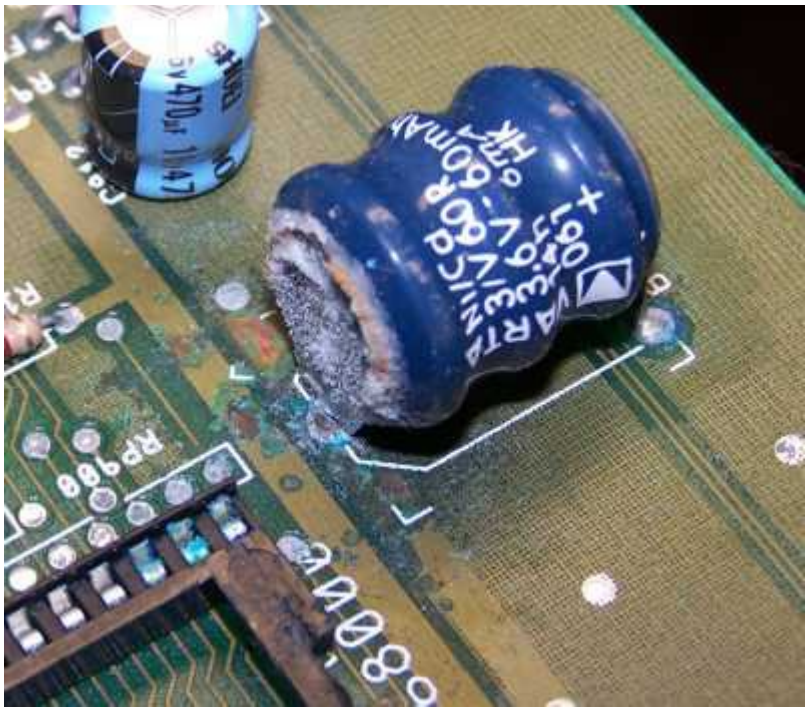
Blinkende LED der Tastatur	
Anzahl Blinks	**Diagnose**
1	Fehler ROM-Püfsumme
2	Fehler im RAM
3	Fehler interner Timer
4	Kurzschluß in der Tastaturmatrix

Amiga Akku Schaden

Hier wird die Reparatur eines Akkuschadens an einem Amiga 2000 in Bildern geschildert. Dank an Joachim Nemetz für diese gut gemachte Anleitung. Vom Prinzip her gilt die Beschreibung für alle derartigen Schäden hervorgerufen durch auslaufende Akku's.

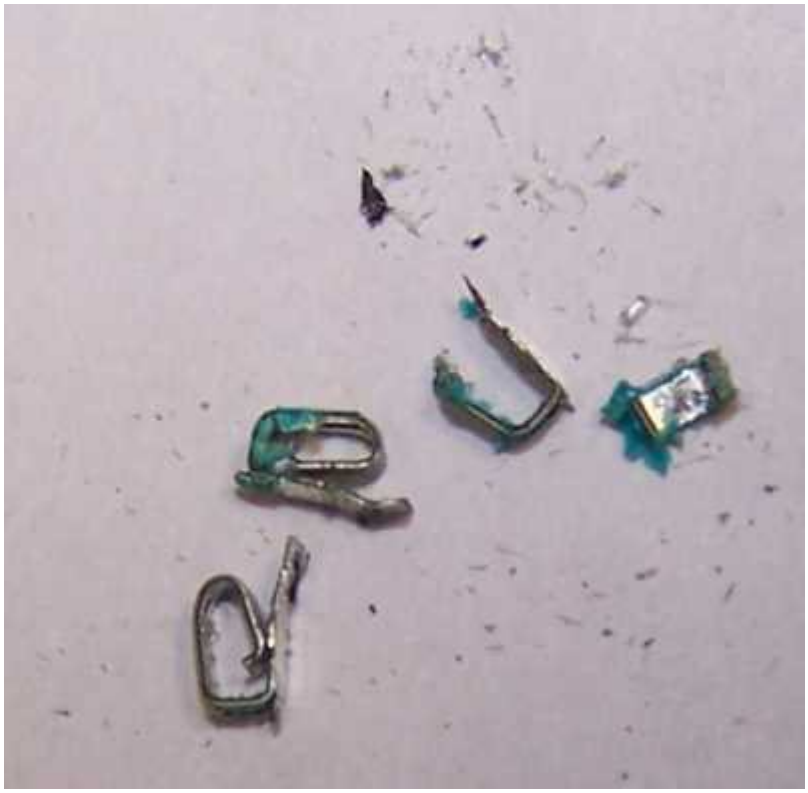
Achtung: Bitte die Akku's regelmäßig kontrollieren und tauschen, bevor es zu solchen Schäden kommt! Am besten auch danach keinen Akku mehr direkt auf der Platine einsetzen, sondern etwas Abseits, sodaß ein späteres Auslaufen zu keinen Schäden mehr an der Elektronik führen kann.

Sichtbare Schäden an der Platine und einigen Pins des 68000er.

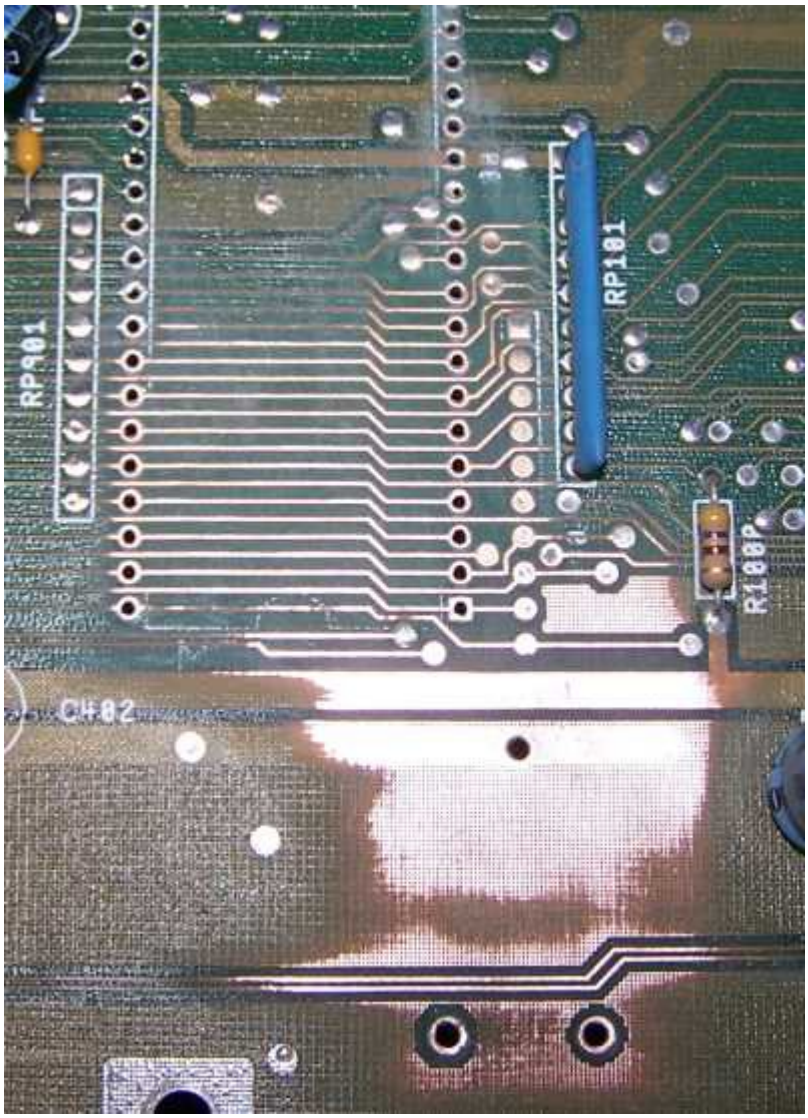


Platine nach Entfernen des Akkus.

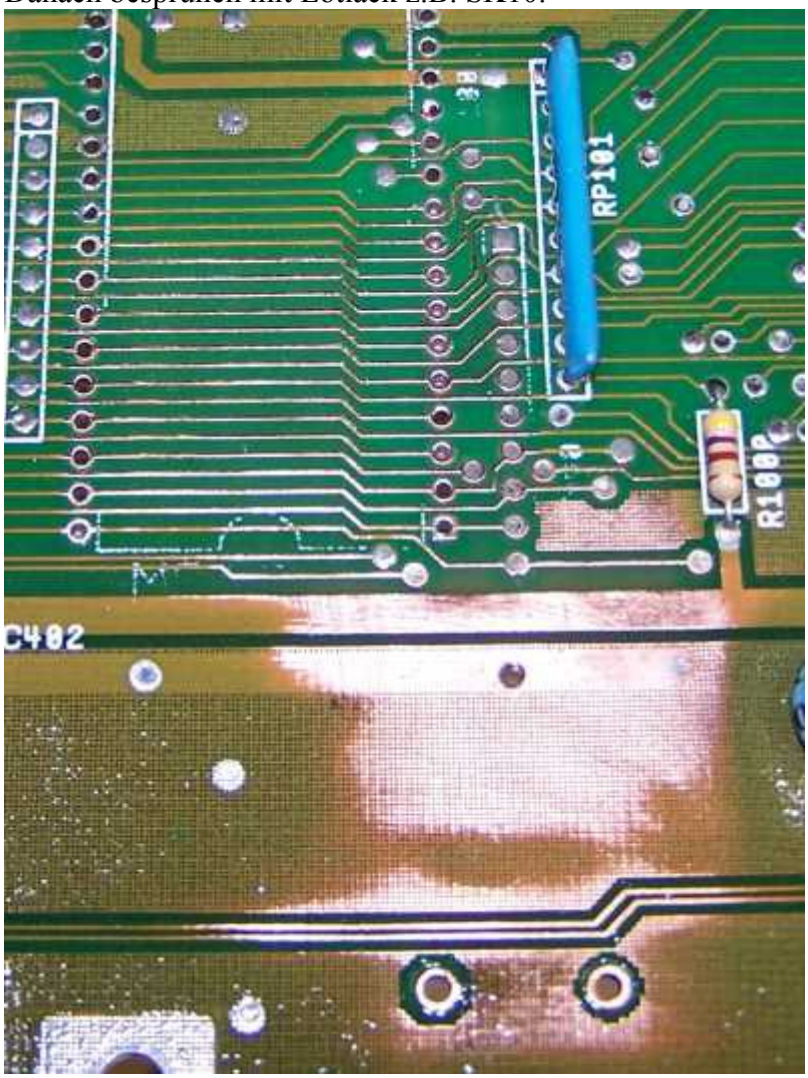
Und das ist das was man in den Pins des Sockels findet. Deshalb immer auch leicht sichtbare Schäden kontrollieren und lieber ein zwei Teile mehr tauschen.



Platine nach dem Abschleifen mit einem Glasfasserstift, danach neutralisieren mit Essig und Nachspülen mit z.B. Spiritus.



Danach besprühen mit Lötlack z.B. SK10.



Sockel und Akku eingelötet, 68000er eingesetzt und.. - läuft wieder.

CF Bootdiskette erstellen

Man braucht dazu folgende Dateien: fat95, CF0 und compactflash.device
Diese Dateien gibt es unter anderem hier:
http://www.forum64.de/wbb3/index.php?page=Thread&threadID=22055

Man nehme eine normale Amiga Diskette und formatiere sie auf 720kb. Unter Windows muss man dazu auf Ausführung gehen und: format a: /t:80 /n:9 eingeben. Dann kopiert man die oben genannten Dateien auf die Diskette.

Jetzt legen wir die Diskette in den Amiga und starten crossdos. Alle 3 Dateien ersteinmal ins Ramdrive kopieren. Diskette rausnehmen und Workbench Diskette einlegen. Evtl. muß man noch etwas Platz schaffen, um die 3 Dateien aus dem Ramdrive jetzt auf die Diskette zu kopieren:

- fat95 ins verzeichnis L
- compactflash.device ins Verzeichnis Devs
- CF0 einfach ins Stammverzeichnis

Jetzt sollte auch von der WB-Disk nach dem Neustart das CF0 verfügbar sein. Es sollte bei eingesteckter CF Karte, automatisch ein Laufwerks-Icon auf dem Desktop erscheinen.

ADF auf Floppy schreiben

Dazu das Programm adf2disk und die zu schreibenden adf Dateien auf die CF Karte kopieren, diese in den PCMCIA Adapter und diesen in den A600 stecken. Mit vorbereiteter CF+WB Bootdisk booten. Doppelklick auf CF Karten Icon. Alle Dateien anzeigen lassen!

Bootdisk entnehmen und zu beschreibende Floppy einlegen. Doppelklick auf adf2disk. In der Eingabezeile ein Leerschritt und dann den zu schreibenden adf Dateinamen eingeben (die auch auf der CF Karte sein sollten). Schon wird das Image auf die Diskette geschrieben.

PS/2 Maus am Amiga (und Atari ST)

Tom Kirk hat folgende Adapterschaltung entworfen, um Standard PS/2 Mäuse am Amiga oder Atari ST anschließen zu können. Die Schaltung besteht nur aus einem PIC 16F84 und wenigen weiteren Bauteilen. Ein Aufbau auf Lochraster sollte also problemlos möglich sein.

Schaltung:

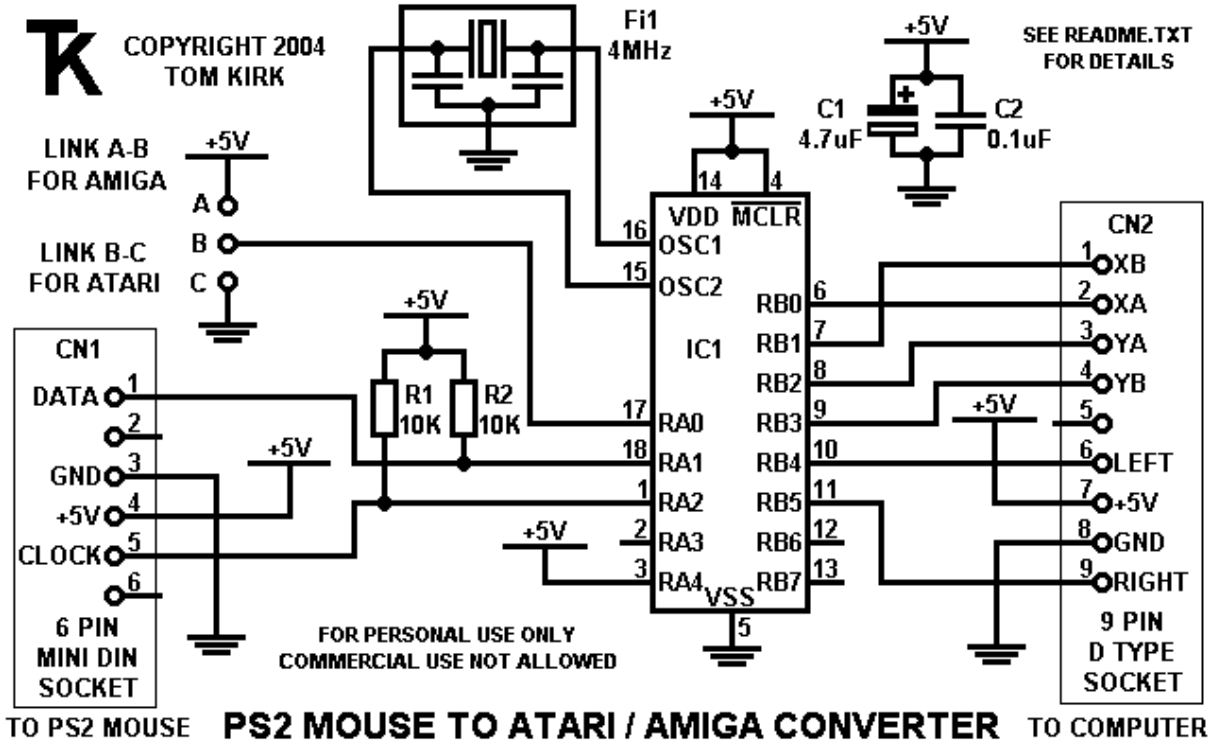
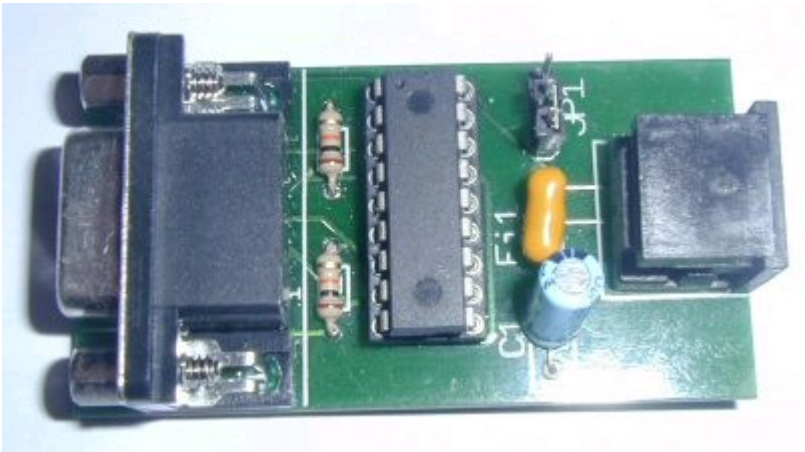


Bild Platinenaufbau:



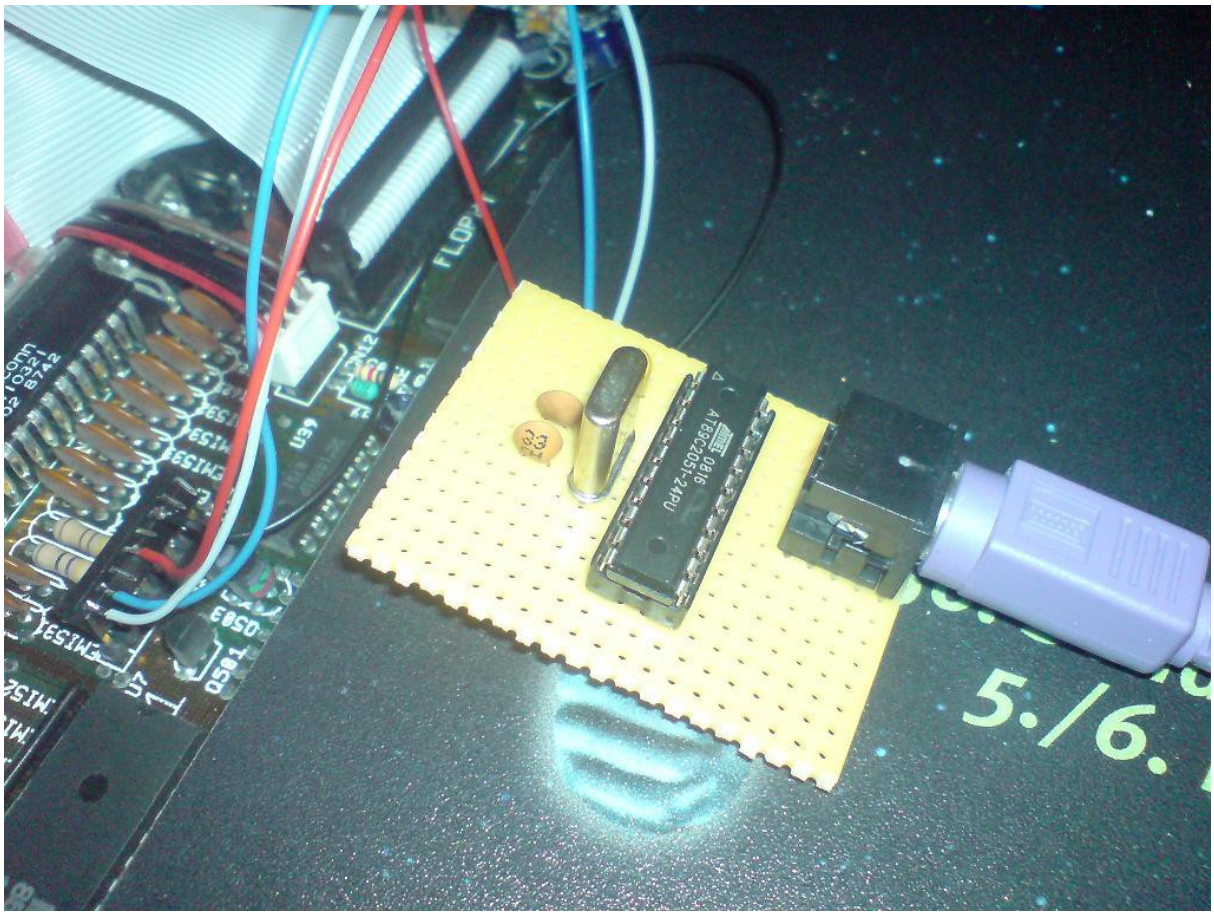
Mehr Informationen und die Firmware des PIC 16F84 gibt es auf Tom's Homepage:
http://members.aol.com/tgkirk/ps2mous.htm

PS/2 Tastaturadapter – nicht nur für A500

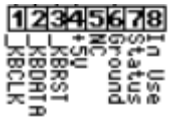
Links:
http://www.rotgradpsi.de/mc/gadget/amiga.html

Auf der Seite von Martin Clausen fand ich die Informationen um mittels eine AT89C2051 Chips einen PS/2 zu Amiga Tastaturadapter zu bauen. Martin war auch so freundlich, mir Programmierte Chips und Quarze zur Verfügung zu stellen. Leider fand ich erst nach der 3ten Auflage dieses Buches, die Zeit, den Adapter einmal auf Lochraster nachzubauen. Das Ergebnis: Der Adapter arbeitet prima!

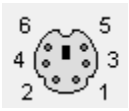
Bild meines Lochrasteraufbaus:



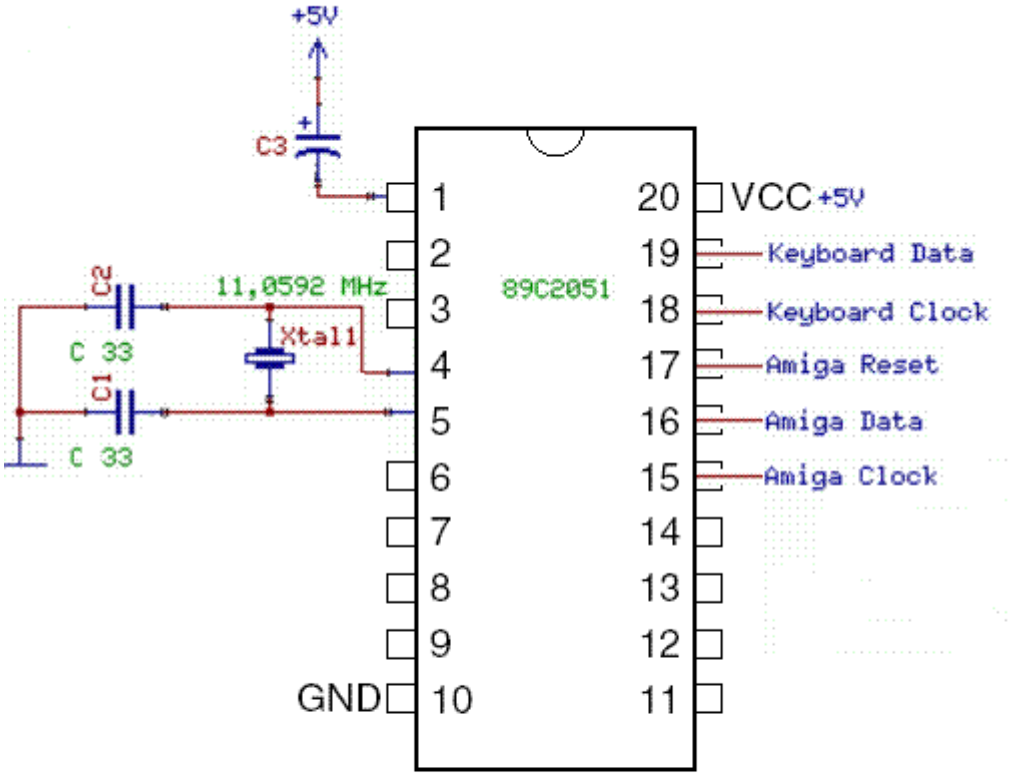
A500 Tastaturschnittstelle:



PS/2 Buchse (1=DATA; 3=GND; 4=+5V; 5=CLK):



Schaltplan:
+5V an Pin20 und GND and Pin 10 des 89C2051 nicht vergessen. Zw. +5V und GND noch einen 100nf Kondensator schalten. C3 ist 1-2,2uF. C1+C2 sind 33pf.

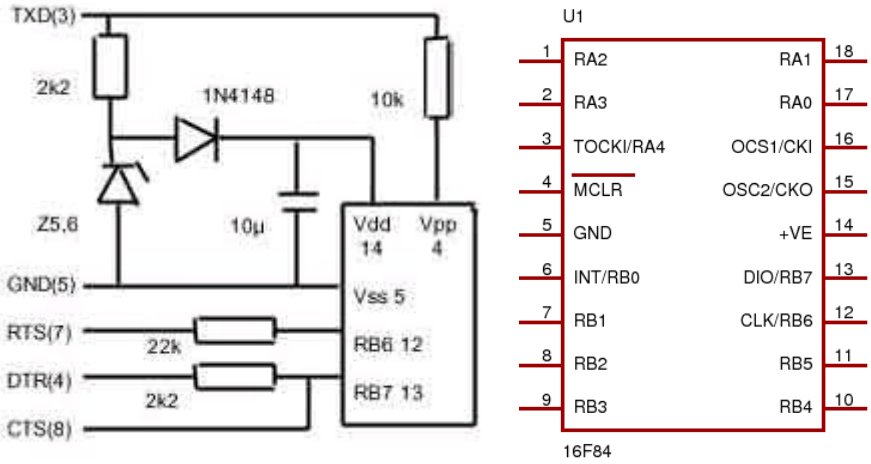


Einfachst PIC Programmer (Ludipipo)

Die Idee zu dieser Schaltung stammt von Erik Herman und wurde zum ersten mal am 12.12.1995 von Ludwig Chatta veröffentlicht. Ludi = Kosename für Ludwig; pi = pic; po = Programmer. So entstand der Name Ludipipo. Mittlerweile gibt es zahlreiche Varianten des Ludipipo oder auch JDM-Programmer. Der JDM Programmer wurde entwickelt von Jens Dyekjær Madsen

Diese einfache Schaltung arbeitet an einer seriellen Schnittstelle eines PC's und benötigt keine externe Spannungsversorgung. Die Pin Nummern in Klammern sind die Anschlüsse einer Sub-D 9-poligen Buchse. An Notebooks kann es zu Problemen kommen, wenn die Spannungen am seriellen Port zu niedrig sind.

Geeignet für alle 18-poligen PIC's (16F84/A, 16F627/8/A etc.). Als Programmiersoftware eignet sich IC-Prog und alle die einen JDM kompatiblen Programmer unterstützen.



Generell empfehle ich den Blick in ein aktuelles Online Auktionshaus mit der Suche nach ‚JDM Programmer' da kann man für ca. 10€ fertige Geräte kaufen.

PC Floppy für Amiga umbauen

Von Stefan Pitsch, ^TePe^ und Stefan Hohendorf.

Unterschiede zwischen PC und Amiga Laufwerken:

1. Das DISK_CHANGE Signal von Pin34 entfernen
2. Das DISK_CHANGE Signal auf Pin2 legen
3. Das READY Signal auf Pin 34 legen
4. Die Disk ID von DS1 auf DS0 ändern (meist über Jumper/Lötbrücken)

D.h. maximal zwei Kabel und eine Lötbrücke.
DISK_CHANGE ist das Signal des Diskwechsels, gerne auch /DC oder DCH abgekürzt.
READY wird gerne auch /RY, RDY o.ä. gekennzeichnet.
DS1 und DS0 werden gerne auch mal nur S1 bzw. S0 genannt.
Das Problem ist nun, bei jeder Platine erstmal herauszufinden, wo diese Signale liegen.

Der Amiga braucht an sich das READY Signal nicht, allerdings gibt es einige Programme (z.B. XCOPY) welche es auslesen und ohne das Signal den weiteren Betrieb verweigern, weil sie meinen, es läge gar keine Diskette im Laufwerk.

Floppy Typen, die erfolgreich umgebaut werden können:

=> http://www.pitsch.de/stuff/amiga/floppy.htm
Sony MPF 920-E
ALPS DF354H127F (oder DF354H090F)
NEC FD1231H
MITSUMI / NEWTRONICS D359T3
TEAC FD235HF 3240-U5
(TePe)
Chinon FB 357
ALPS(IBM) DF354H021E (no Disk Change)
EPSON SMD 300
Mitsubishi MF355-3250MG
Panasonic JU-257A427P
Samsung SFD-321B
Samsung SFD-321B EF
Samsung SFD-321B KFUJ
SONY MPF 920 E
SONY MPF 920 V
Y-E Data Typ 1
Y-E Data Typ 2
TEAC FD235HF-217-U
TEAC FD235HF-3240-U5
TEAC FD235HF-4291-U
TEAC FD235HF-5240
TEAC FD235HF-6291

TEAC FD235HF-7291
Mitsumi D359M3 (Typ 1)
Mitsumi D359M3 (Typ 2)
Mitsumi D359M3D
Mitsumi D359T2
Mitsumi D359T3 (Typ 1)
Mitsumi D359T3 (Typ 2)
Mitsumi D359T5 (Typ 1)
Mitsumi D359T5 (Typ 2)
Mitsumi D359T6 (Typ 1)
Mitsumi D359T6 (Typ 2)
Chinon FZ357
EPSON SMD-300
Mitsubishi MF355F-3250MG
Mitsumi (Newtronics) D359T5 und D359T6
Panasonic JU-257A604P und JU-257A656P
Teac FD-235HF 218U, 4430, 4291, 5291-U, 6291 und 7291-U5
Y-E Data 702D
Mitsubishi MF355F-3250MG (anderes Board)
MITSUMI 5K0800G und 3J122JA
Mitsumi D359M3 / D359T3
Sony MPF 920-E
NEC FD1231T / FD1231H
SAMSUNG SFD-321B

Beispiel: NEC FD1231H

Auch hier gibt es wieder mehrere Boardvarianten. Meines sah z.B. aus wie ein FD1231T und passte daher nicht zu der Anleitung aus dem a1k.org Forum.

Der Aufbau:

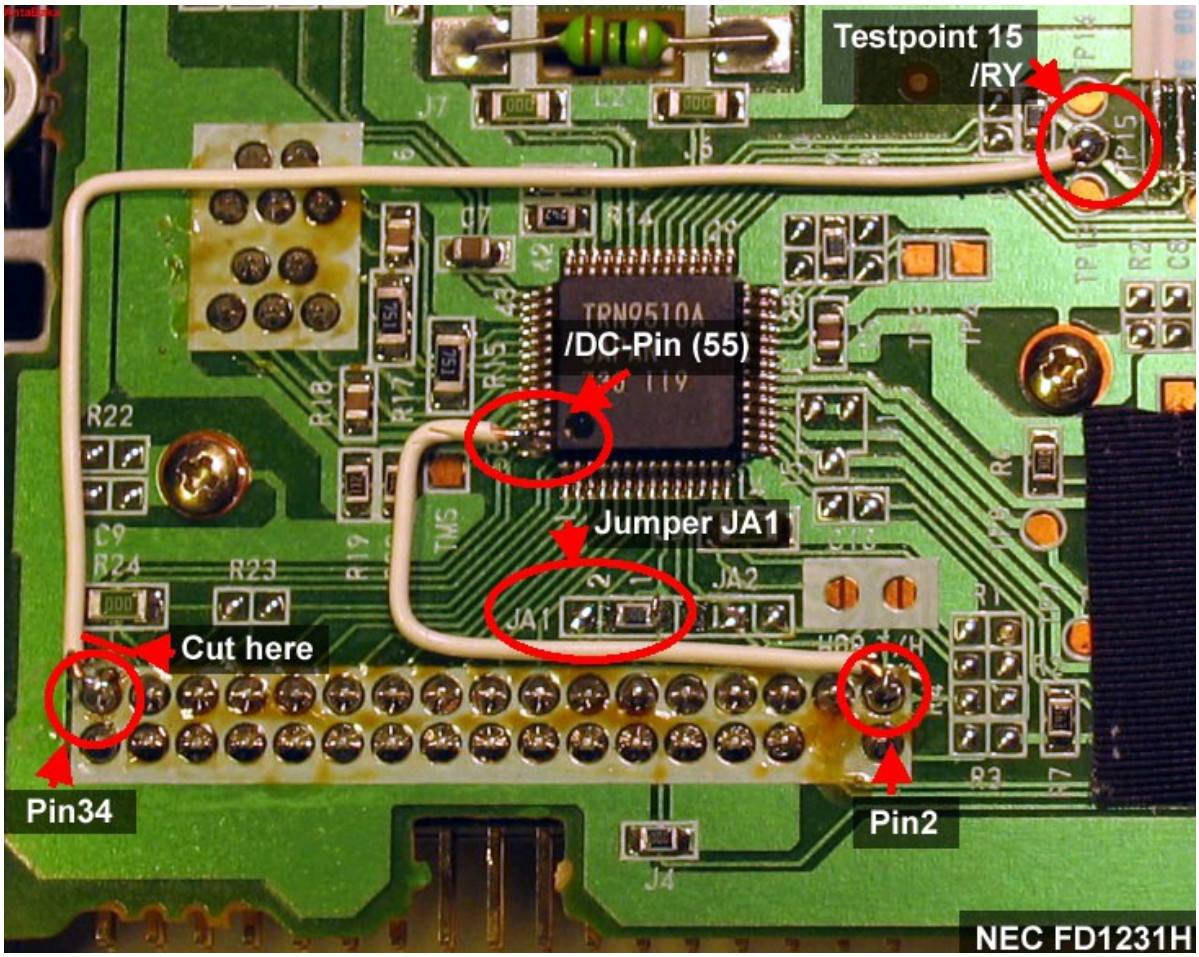
1. Zunächst verlegt man am Jumper JA1 (mittig über der Steckerleiste) den 0 Ohm Wiederstand von der Stellung 2 nach rechts auf die Stellung 1.

2. Dann muss man oberhalb des Pin34 die Leiterbahn durchtrennen (auskratzen).

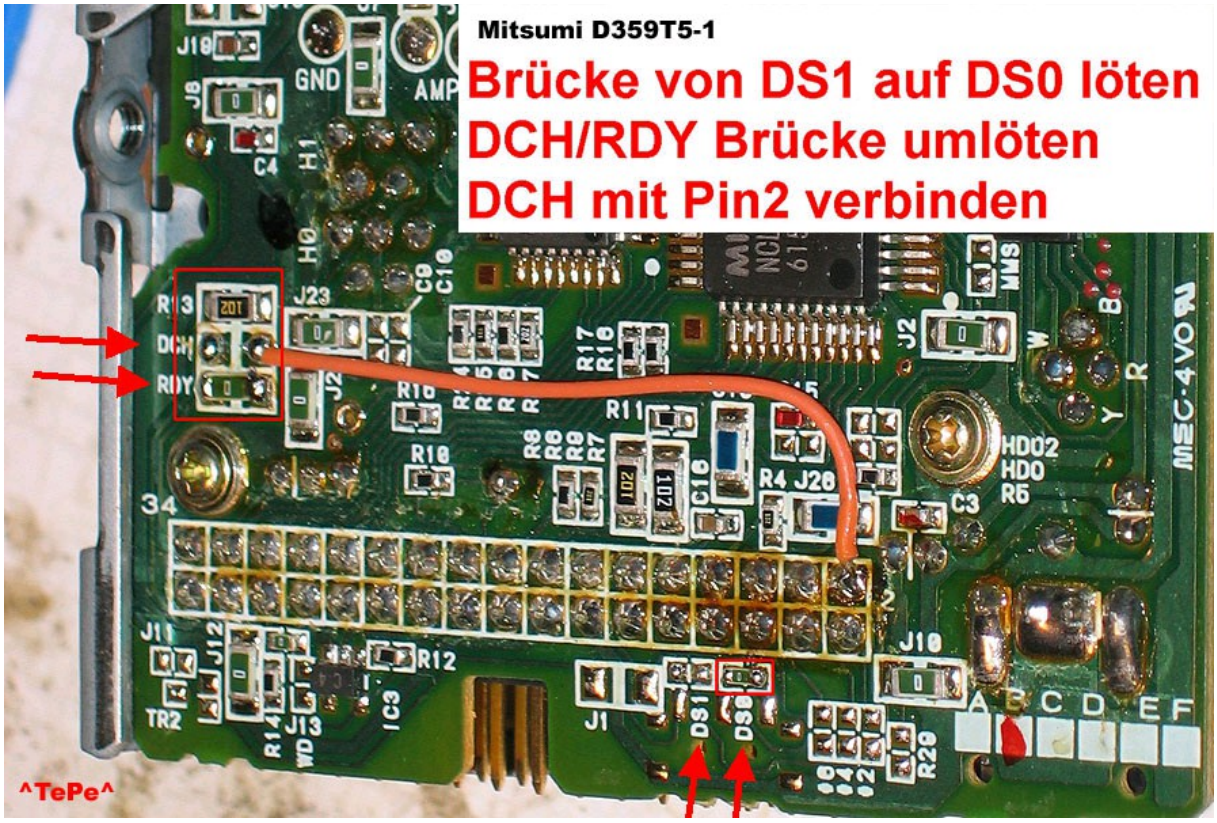
3. Jetzt wird eine Verbindung zwischen dem TP15 und Pin34 geschaffen - das ist /RY.

4. Nun muss man noch von der Leiterbahn, die vorher an Pin34 dran war, eine Verbindung zu Pin2 schaffen. Ich habe hierzu direkt an den dazugehörigen IC-Pin (Pin55) angelötet.

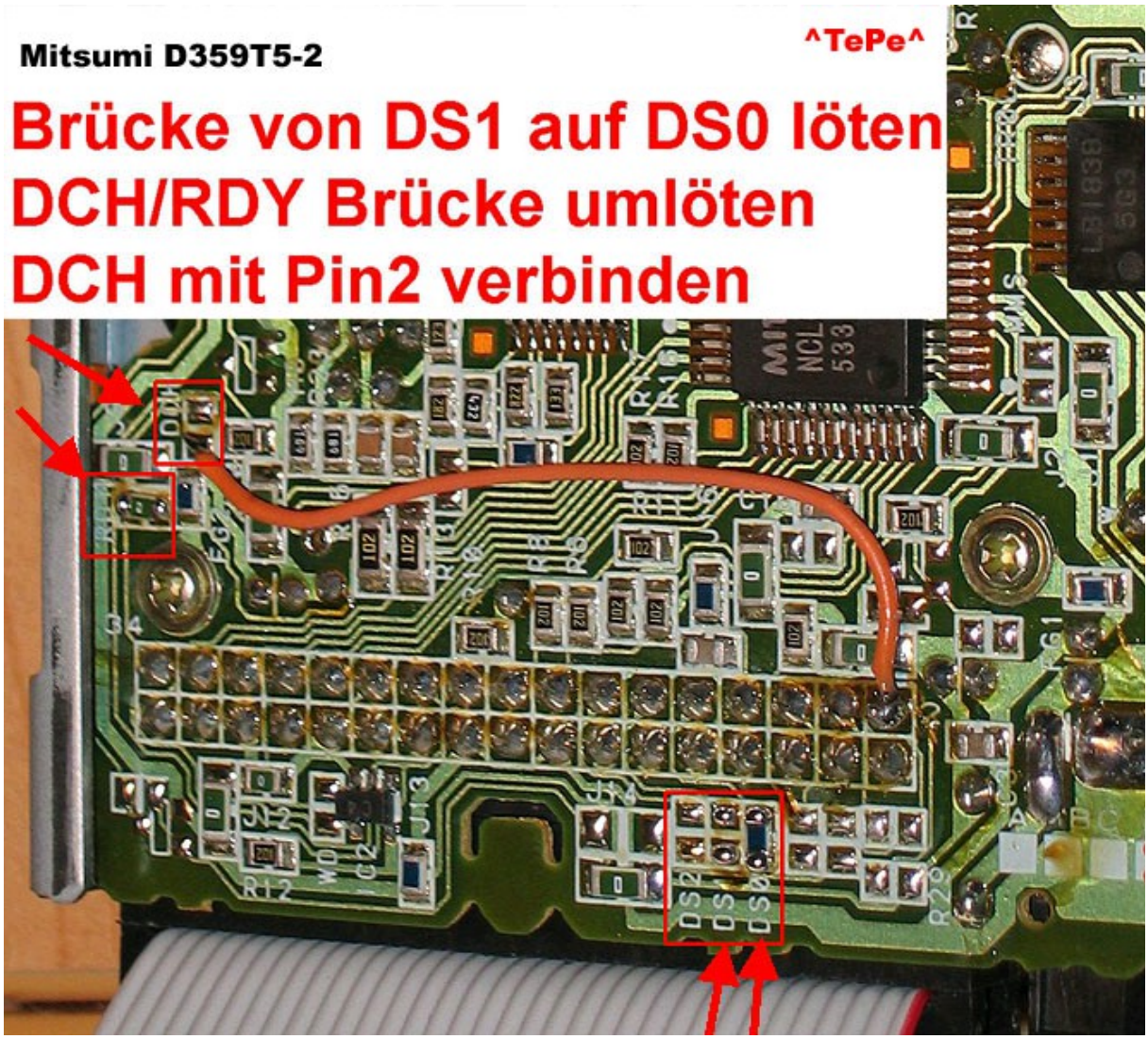
HINWEIS: Ein so umgebautes LW funktioniert nicht ganz korrekt. Solange es das einzige LW am Rechner ist, ist alles OK, aber sobald man eine zweite Floppy dazu anschließt, kommt es zu Problemen - meistens lässt sich die zweite Floppy gar nicht ansprechen. Das liegt daran, dass es sich bei dem /RY-Signal nicht um das echte /RY Signal handelt. Im Zweifelsfalle sollte man also zu einem anderen Umbau greifen (z.B. Mitsumi oder TEAC).



Beispiel: MITSUMI D359T5



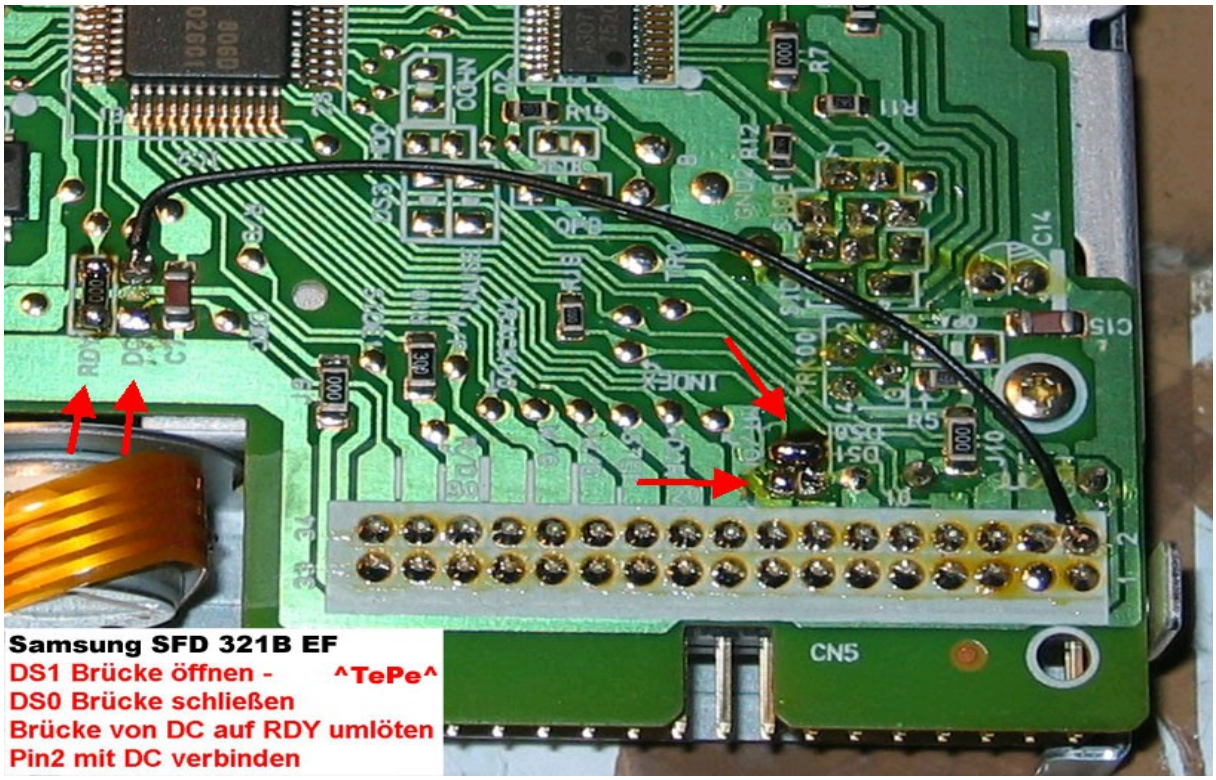
Etwas andere Variante:



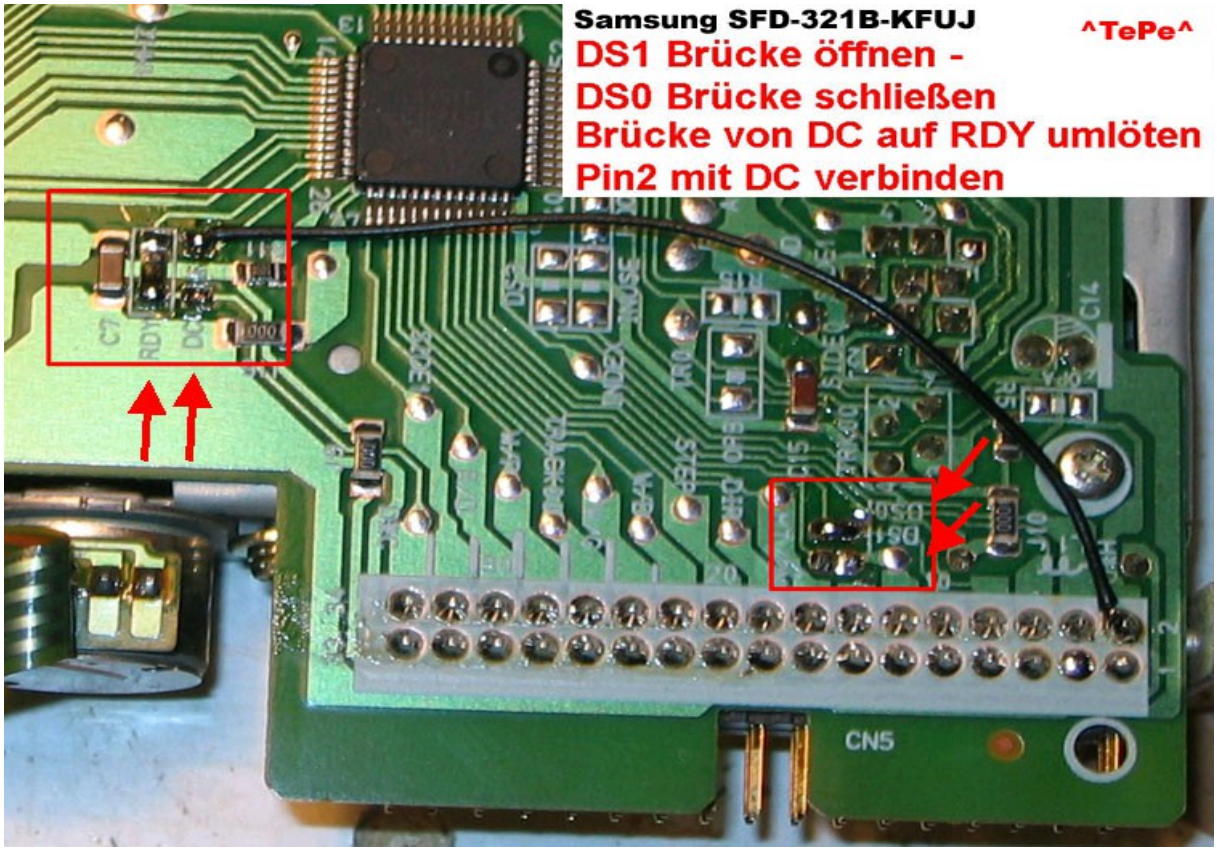
Beispiel: SAMSUNG SFD-321B

Das Öffnen dieses Laufwerks gestaltet sich etwas schwierig:

- Frontblende abnehmen
- unten die 4 Schrauben lösen, die das Gehäuseblech halten
- die eine Schraube lösen, die das obere Gehäuseblech hält
- dann zuerst das obere
- dann das untere Gehäuseblech nach vorne wegziehen


Samsung SFD 321B EF
DS1 Brücke öffnen - ^TePe^
DS0 Brücke schließen
Brücke von DC auf RDY umlöten
Pin2 mit DC verbinden
CN5

Varianten:

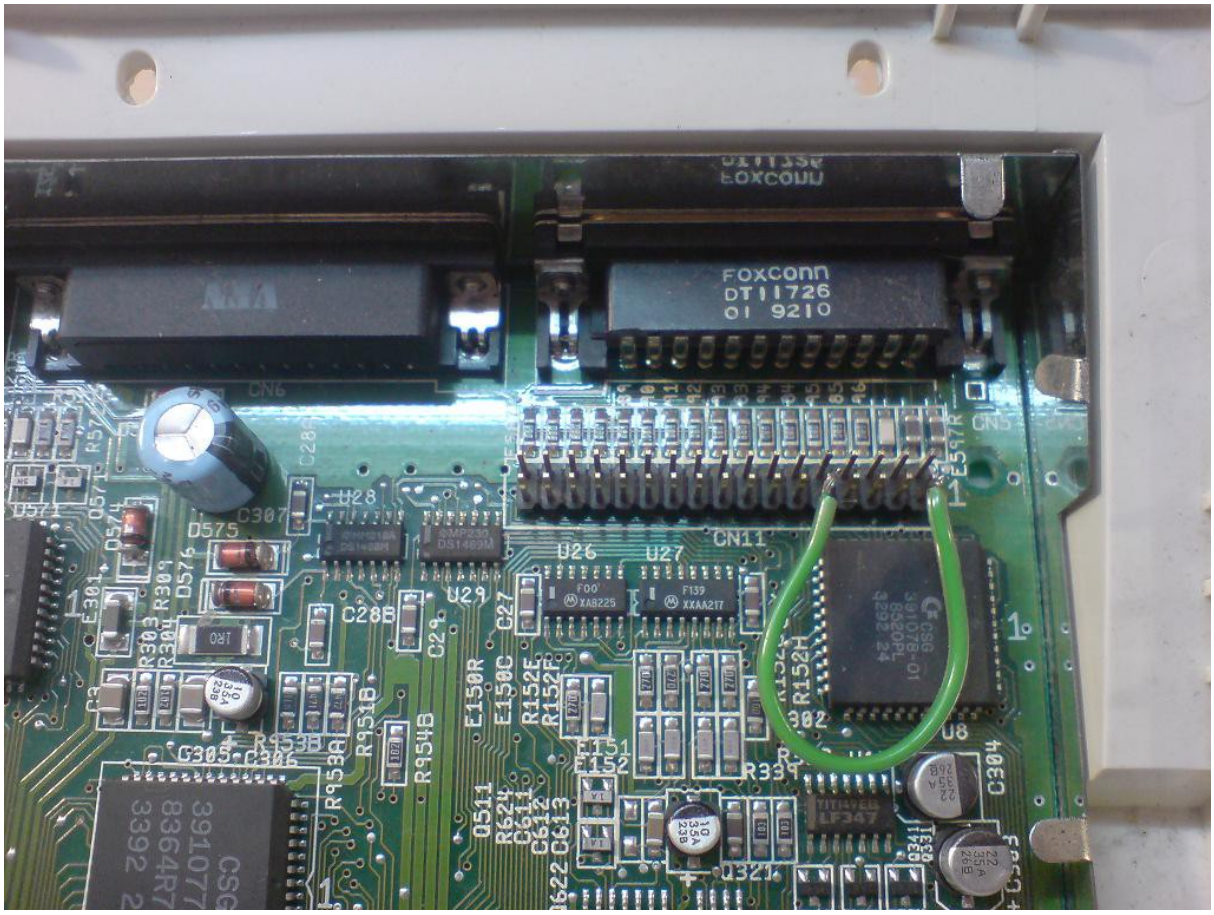

Samsung SFD-321B-KFUJ
^TePe^
DS1 Brücke öffnen -
DS0 Brücke schließen
Brücke von DC auf RDY umlöten
Pin2 mit DC verbinden
CN5

Floppy ‚NO Disk' Hack

Wenn man auf die interne Floppy verzichten möchte, da man z.B eine interne (CF) Festplatte eingrichtet hat und bei A600/1200 Daten mit dem PC über PCMCIA+CF Karte austauscht, so kann man sicher die interne Floppy eingebaut lassen.. aber zum einen ‚schleppt' man sie so immer mit rum und zu anderen greift der Amiga ca. alle 5 Sekunden auf das Laufwerk zu, um zu prüfen ob eine Diskette eingelegt wurde (hört man am leisen klacken..).

Nun, die Floppy einfach ausbauen, spart das Gewicht.. aber nun braucht der Amiga beim booten noch länger.

Um das Warten zu vermeiden, kann man am Floppyanschluß eine Kabelbrücke zw. Pin 2 und Pin 10 (Im Bild in der unteren Reihe sind das Pin 1 und Pin 5 von rechts) einlöten. Dadurch ‚erkennt' der Amiga ein Floppylaufwerk ohne eingelegte Diskette und geht somit direkt zum nächsten Boot-Device, was dann in den meisten Fällen die interne Festplatte sein dürfte. Damit geht das Booten wieder so flott wie mit angeschlossenem Floppylaufwerk.



A500+ auf 2MB Chipram Umbau

Nach jopower basierend auf einer Anleitung von The_Killer

Benötigte Bauteile:

8 Rams 4*256K 120nS oder schneller (am besten der gleiche Typ, wie bereits im Amiga vorhanden).
Isolierte Litze.

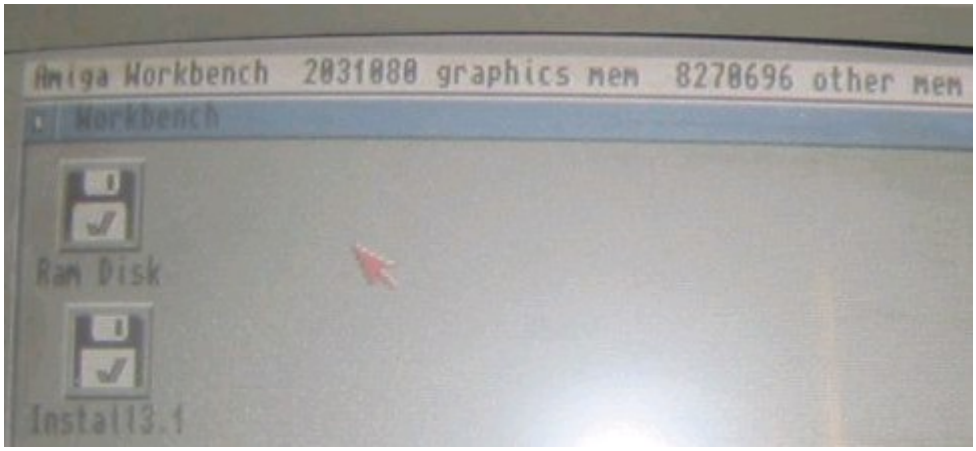
Der A500+ ist bereits für 2MB Chipram vorbereitet, was den Umbau sehr vereinfacht.

Bei den 8 Rams wird Pin 17 nach oben gebogen. Nun werden die Rams huckepack auf die schon vorhandenen gesteckt und alle Pins (bis auf Pin 17) miteinander verlötet. Die aufgelöteten Rams bekommen die gleiche Bezeichnung wie die Rams darunter, nur mit einen "`" am Ende.

Alle Pin 17 von U16`, U17`, U20` und U21` werden miteinander und mit Pin 5 von U32 verbunden.

Alle Pin 17 von U18`, U19`, U22` und U23` werden miteinander und mit Pin 11 von U32 verbunden.

Damit ist der Umbau beendet, und der Amiga verfügt nun über 2 MB Chipram.

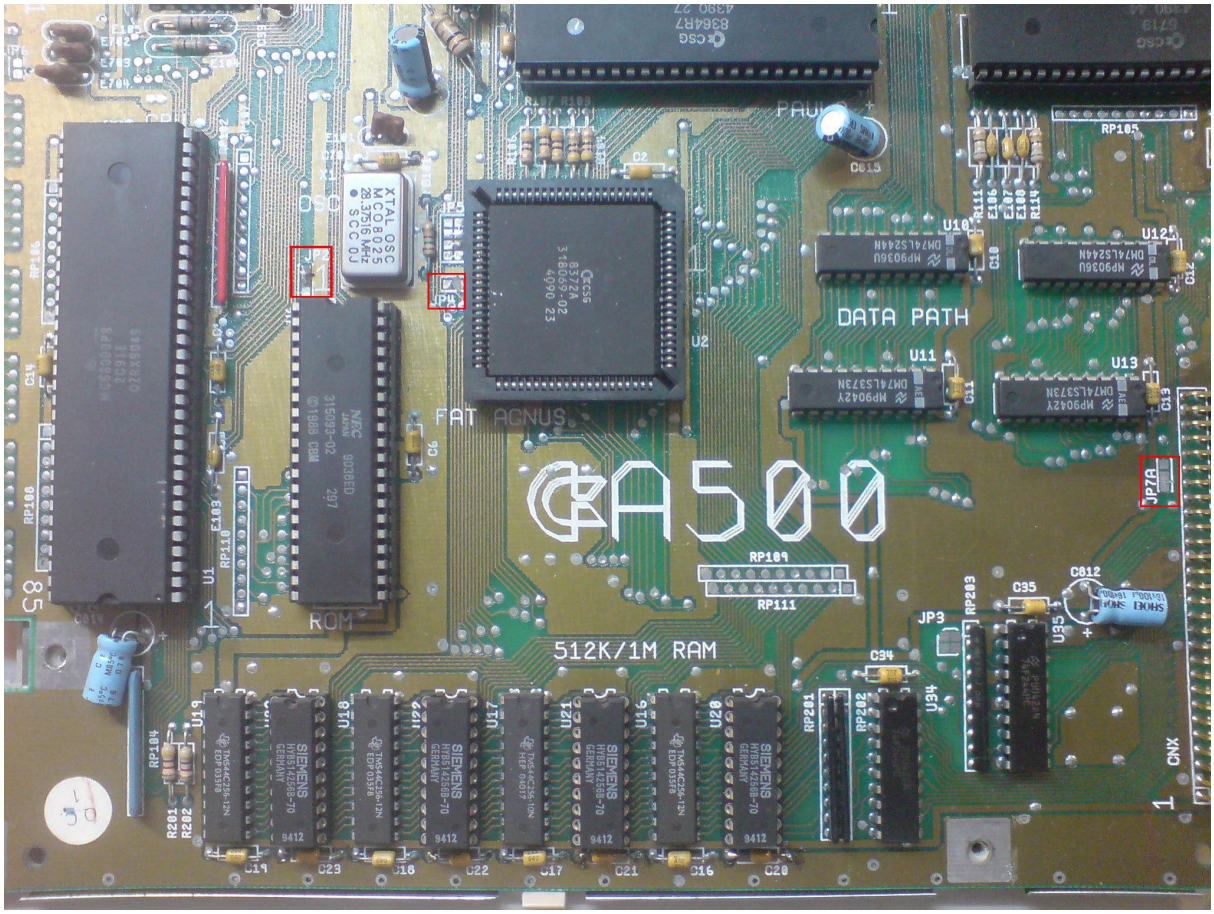


Amiga 500 intern auf 1MB Ram aufrüsten

Board-Rev. 6A

- JP2 von Mitte unten nach Mitte oben ändern
- JP4 auftrennen
- Fat Agnus 8371 gegen Fat Agnus 8372A austauschen (U2)
- Freie RAM-Plätze U21-U23 mit gleichen RAM-Typen nachbestücken (Sockeln)

Achtung: Eine RAM-Erweiterung über die Steckkarte A501 ist dann nicht mehr möglich.



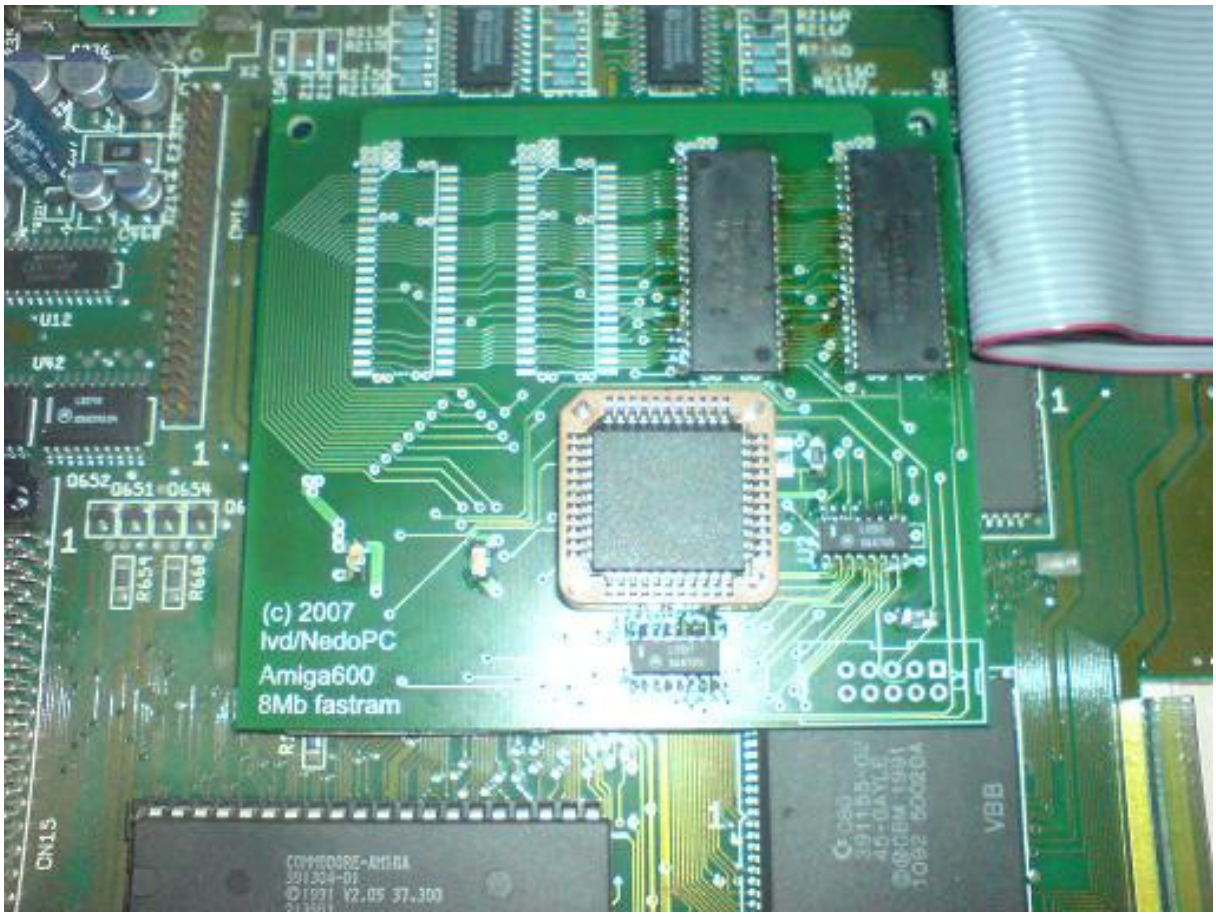
Die Rev. 5 oder 3 lassen sich prinzipiell auch aufrüsten, das Verfahren ist aber komplizierter, weshalb ich hier nur die Erweiterung über eine 512k Ramerweiterung im Erweiterungsschacht (sog. Trapdoor) empfehlen möchte.

A600 Fastramerweiterung 4/8MB

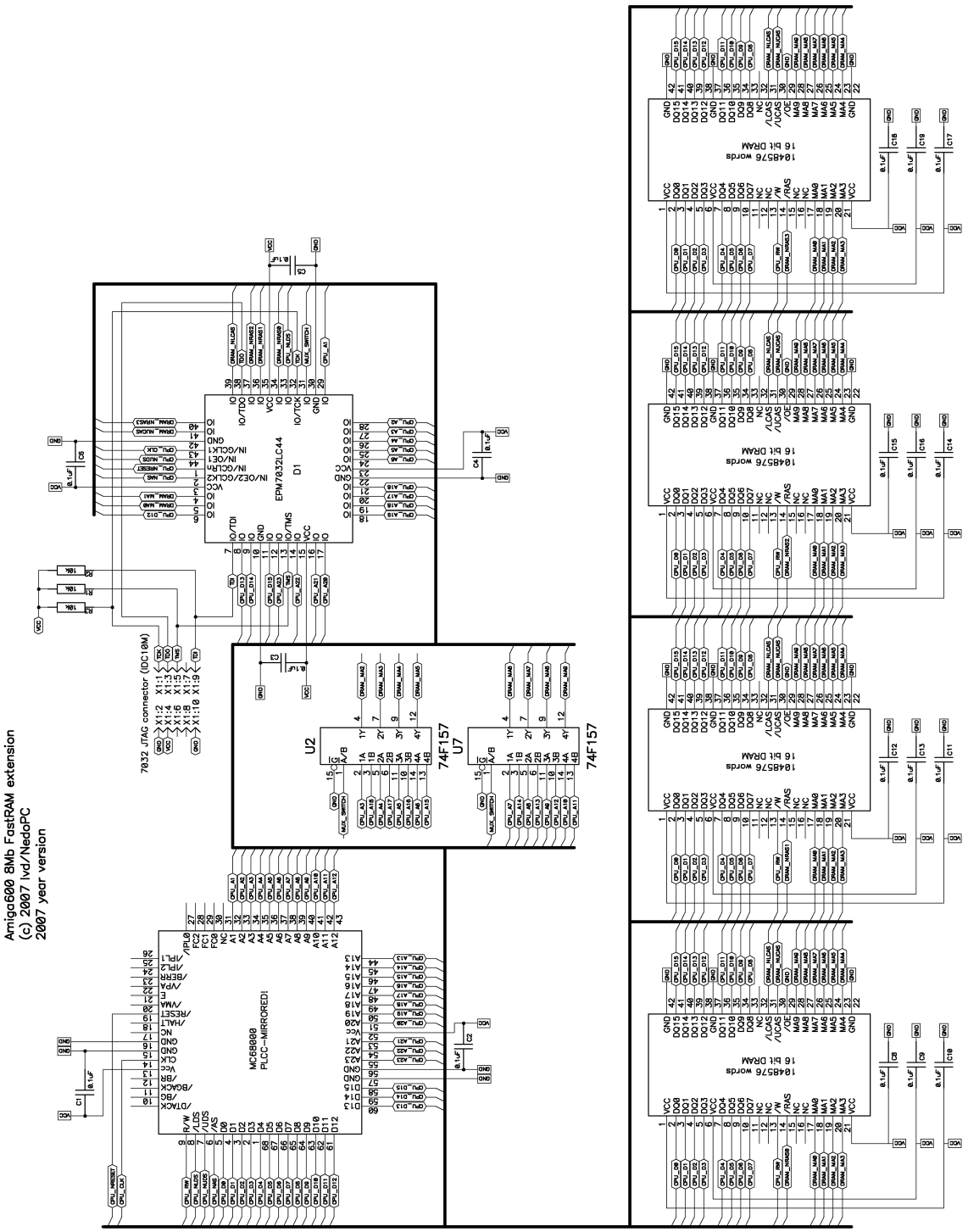
Author: **Vadim Akimov**
Homepage: http://lvd.nedopc.com/Projects/a600_8mb/index.html

Hier wird ein Nachbauprojekt vorgestellt, das 4 oder 8MB (je nach CPLD Konfiguration) Fastram zusätzlich für den Amiga 600 zur Verfügung stellt. Bei der 4MB Version kann die PCMCIA Funktionalität parallel genutzt werden. Bei der 4MB Version werden nur die beiden rechten Ramchips benötigt. Beim Auf-/Einbau zu beachten: 68-pol PLCC Sockel mit mind. 2-3mm Abstand zur Platine auflöten. Quarzoszillator und Quarz auf dem A600 Board mit z.B Isolierband isolieren!

Im a1k.org wurde in 2010 eine Sammelbestellung organisiert, um dieses Projekt nachzubauen. Board-Layout ist in P-CAD 2001. Platinen können z.B bei Conrad angefertigt werden. Die zentrale Logik ist in einem CPLD EPM7032SLC44-10 oder ATF1502AS10JC44 untergebracht. Ram Chips sind 1 Megaword 16-BIT 42-Pin's SOJ Dram's mit <= 60-70ns. Z.B: NN5118160AJ-60, GM71C18160AJ7, TMS418160***-60 etc. Desweiteren werden 2 x 74F157 SMD IC's, 3 x 10k 1206 SMD Widerstand und 2 + 3 x je Ram 100nF 1206 SMD Abblockkondensatoren benötigt.



Schaltplan:

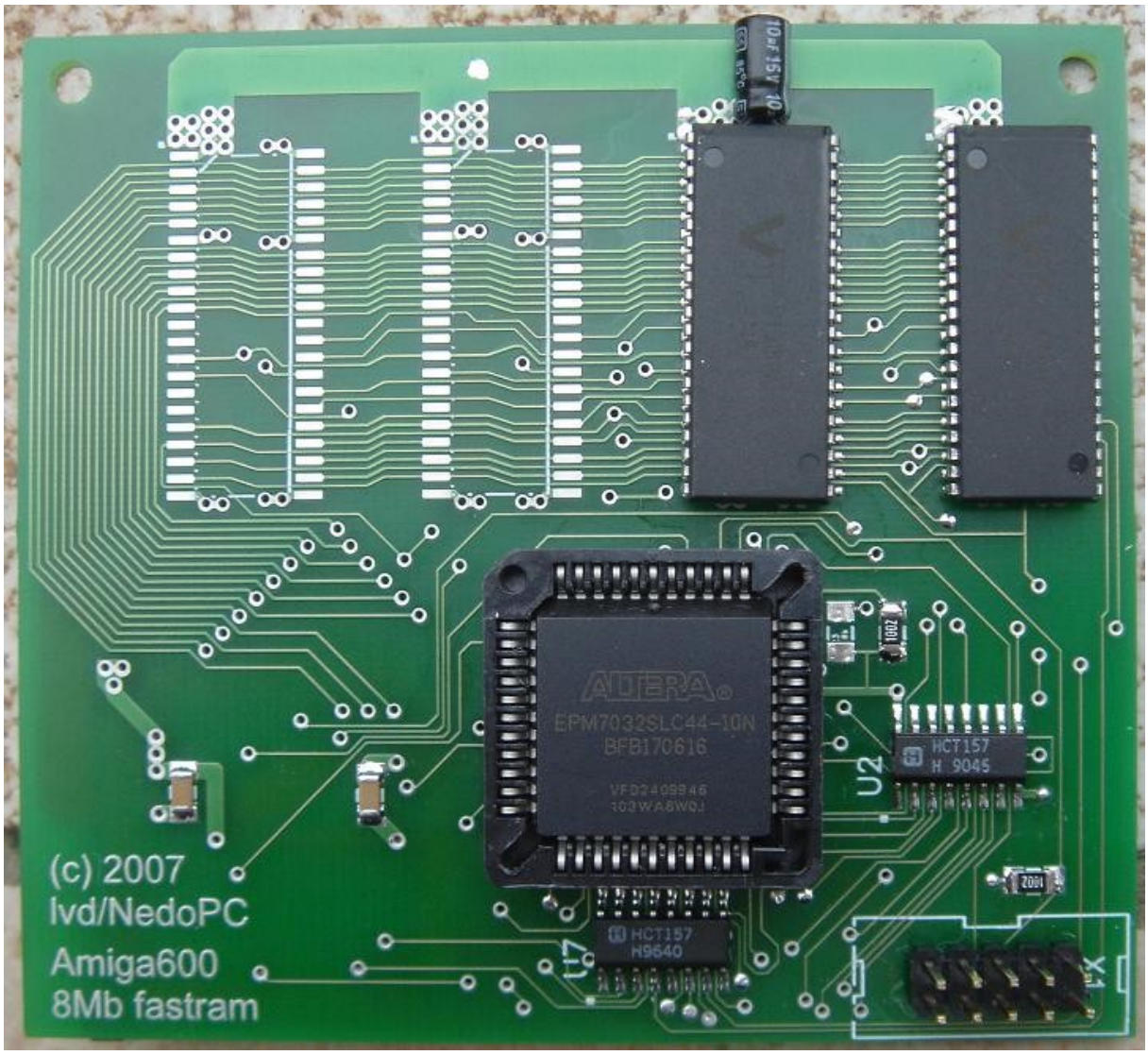

Amiga600 8Mb FastRAM extension
(c) 2007 ivd/NedoPC
2007 year version
7032 JTAG connector (IDC10M)
EPM7032LC44
MC68000
PLCC-MIRRORED!
74F157
16 bit DRAM
1048576 words

Verbesserungsvorschläge (Dank an botfixer aus dem a1k Forum):

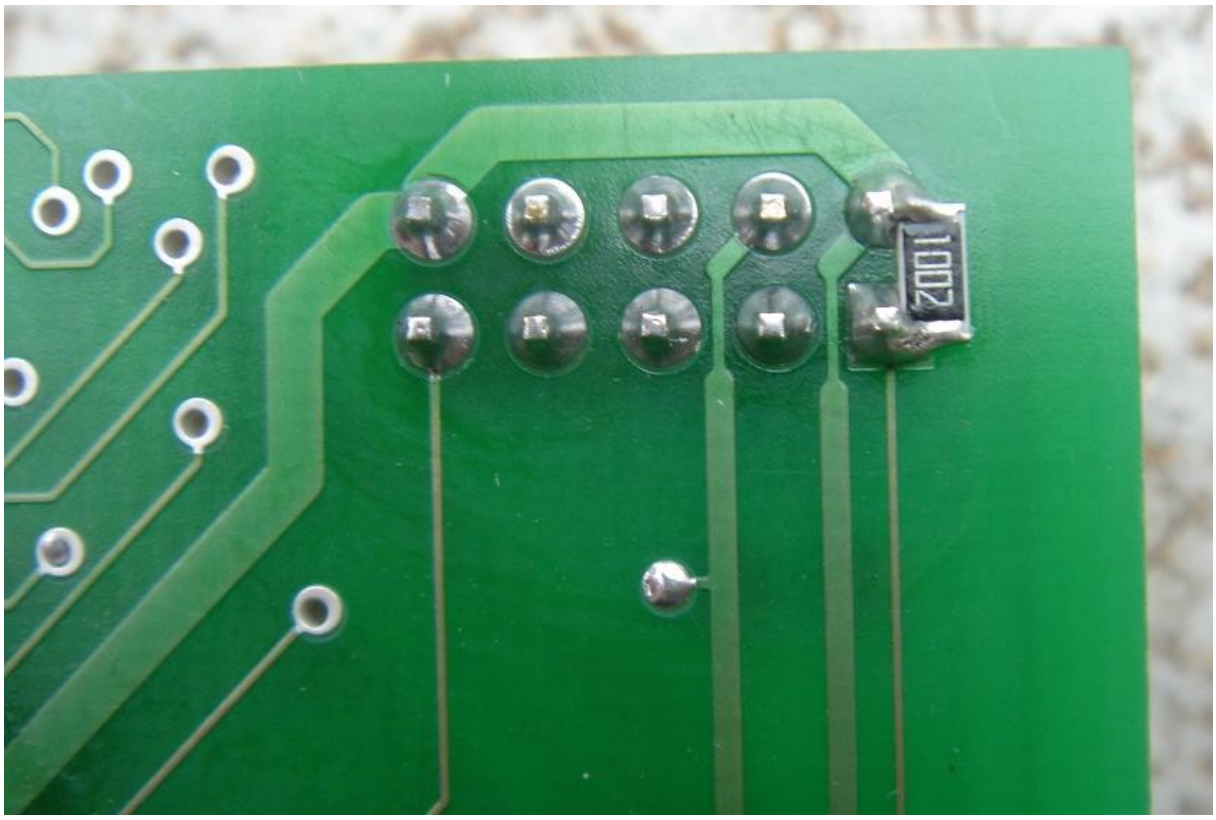
Hier sieht man eine 8MB Platine. Diese ist so zusammengebaut wie es in der originalen Dokumentation beschrieben ist. Die Elkos sind auf beiden Platinenseiten verlötet und machen es erforderlich, den Platinenlack anzukratzen um dort anzulöten. Das ist aber völlig unnötig, weil man die Elkos (die hier gezeigten Subminiaturelkos) oben montieren kann. Das PCB hat dazu sogar mehrere gut positionierte VIAs, wo man die Elkodrähte einstecken und verlöten kann.

Des weiteren ist für eine betriebssichere Funktion des JTAG Connector ein Pull-Down Widerstand notwendig. Das PCB hat einen Fehler, so dass man diesen Pull-Down Widerstand mit einem Stück Draht nach Masse verbinden muss, so wie auf dem Bild zu sehen:

Ich würde für die Elkos aber folgendes Vorschlagen (Minuspol rechts):



Das ist eine weitere funktionsfähige Platine (4MB), wo ich ein Elko als Beispiel mal so eingesetzt habe.



Das hier ist die Rückseite vom JTAG Connector. Wenn man den Widerstand so dort einlötet, hat es die gleiche Funktion und Verbindung als wenn man das ganze wie auf der anderen Karte quer mit einem Kabel verdrahten muss.

Ich finde, die beiden Verbesserungen sehen viel besser aus und lassen sich auch viel leichter umsetzen.

Floppy Emulatoren

Hier Links zu entsprechenden Projekten:

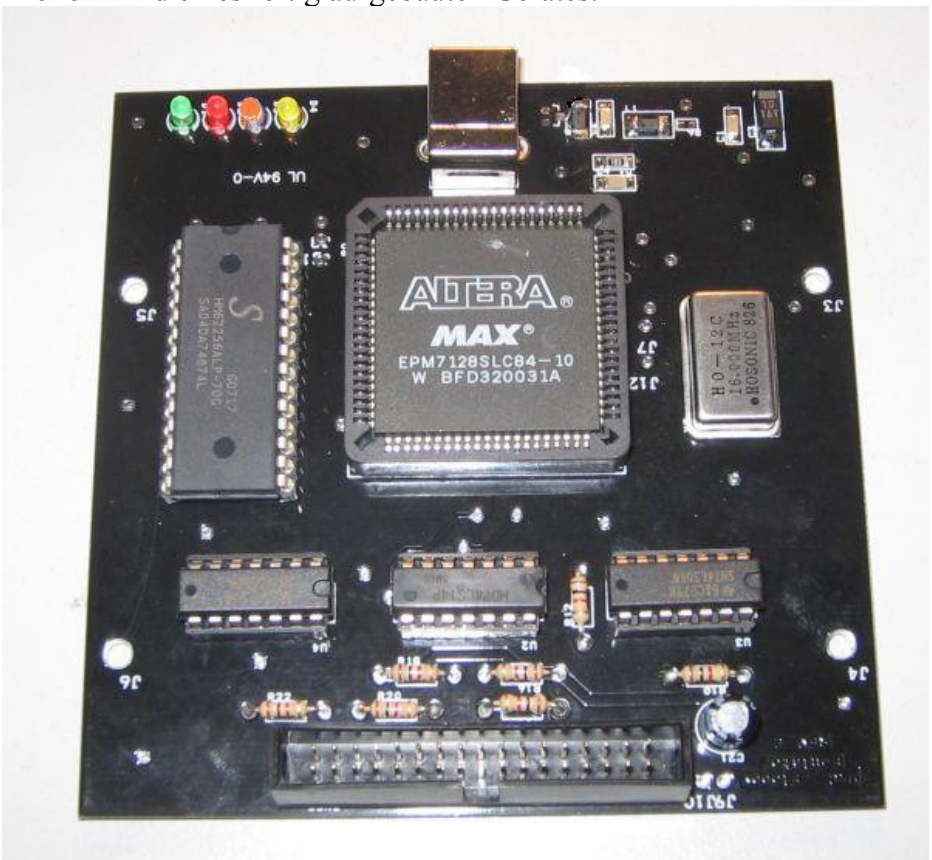
http://www.torlus.com/floppy/
http://jeanfrancoisdelnero.free.fr/floppy_drive_emulator/index.html
http://www.thesvd.com/SVD/
http://atariamiga.free.fr/sdiskemul.php
http://www.sensi.org/~tnt23/megadrive/

Bei vielen Projekten ist leider keine allzu große Aktivität festzustellen. Einzig das HxC USB Projekt hatte alle Voraussetzungen:

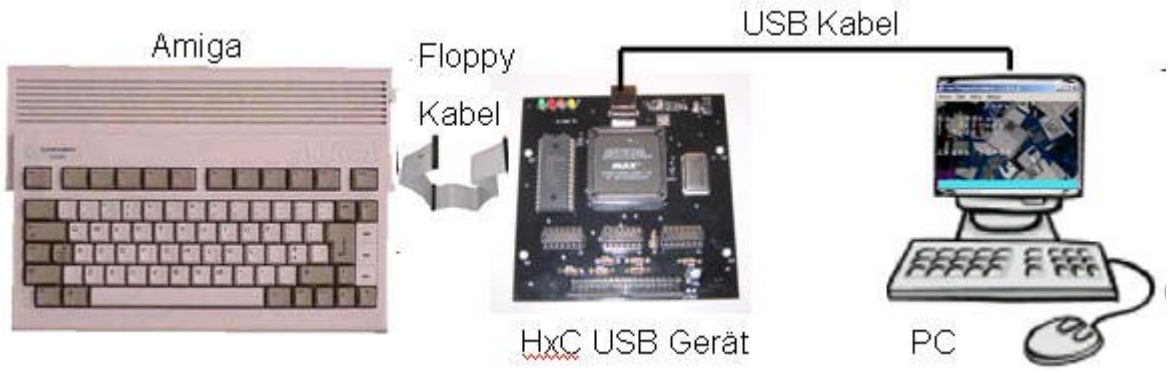
- Platinenlayout als Gerber/Eagle verfügbar
- Firmware verfügbar
- Schaltplan verfügbar
- Software verfügbar
- Getestet (20 Einheiten wurden gefertigt)
- Aktiv!

Nach langem hin- und herüberlegen, habe ich mich dann entschlossen im forum64, a1k und CPC-forum nach Interessenten Ausschau zu halten. Letzlich ist dann eine Bestellung über 20 Platinen (pcbcart.com) entstanden. Über das a1k Forum meldete sich auch noch jemand, der noch einen Restposten passender CPLD's hatte und uns diese sogar programmiert zu einem guten Preis zur Verfügung stellen konnte. Insgesamt sind mit ca. 50€/Stück zu rechnen.

Hier ein Bild eines fertig aufgebauten Gerätes:



Und so wird das Gerät zwischen Amiga und PC/Notebook verkabelt:



Bedeutung der LED's:

D1 – Power on (Strom kommt vom USB Anschluß
D2 – Floppy Aktivität (Wenn der Zielcomputer auf die Floppy zugreift)
D3 – Step (Wenn der virtuelle Lesekopf bewegt wird)
D4 – USB Link

Noch ein paar Tips zum Aufbau und Test:

D5 kann weggelassen werden.
L3 mit Drahtbrücke überbrücken
R6 ist 0 Ohm oder Drahtbrücke
Zuerst messen, das zw. 5V und Masse kein Kurzschluß ist.
Dann das Gerät über USB an den PC anschließen (Jetzt muß der FT245 Treiber installiert werden). Dann die HxC PC Software starten. Es sollte dann die Meldung kommen, das das Interface gefunden wurde! Unter USB Stats Menüpunkt sollte ein Durchsatz von ca. 32768 Byte/Sekunde zu sehen sein.

Dann das USB Kabel wieder trennen, die Verbindung Zielcomputer und Gerät über Floppykabel herstellen, USB Kabel wieder einstecken und ein Floppyimage einfach per Drag und Drop auf das Fenster des PC Programms ziehen..

Ein wirklich schönes Stück Hardware.

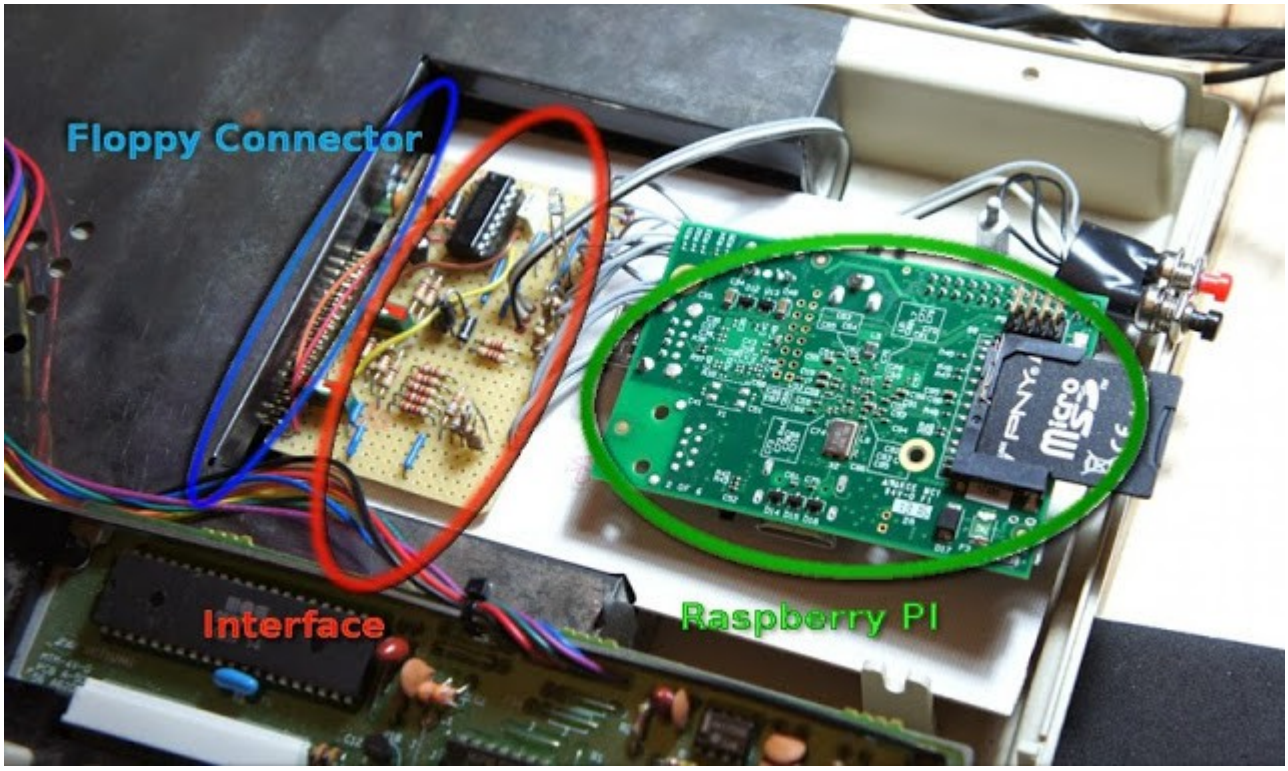
Amiga RPi Floppy Emulator

Link:
http://amigadrive.blogspot.it/
http://www.a1k.org/forum/showthread.php?p=687008

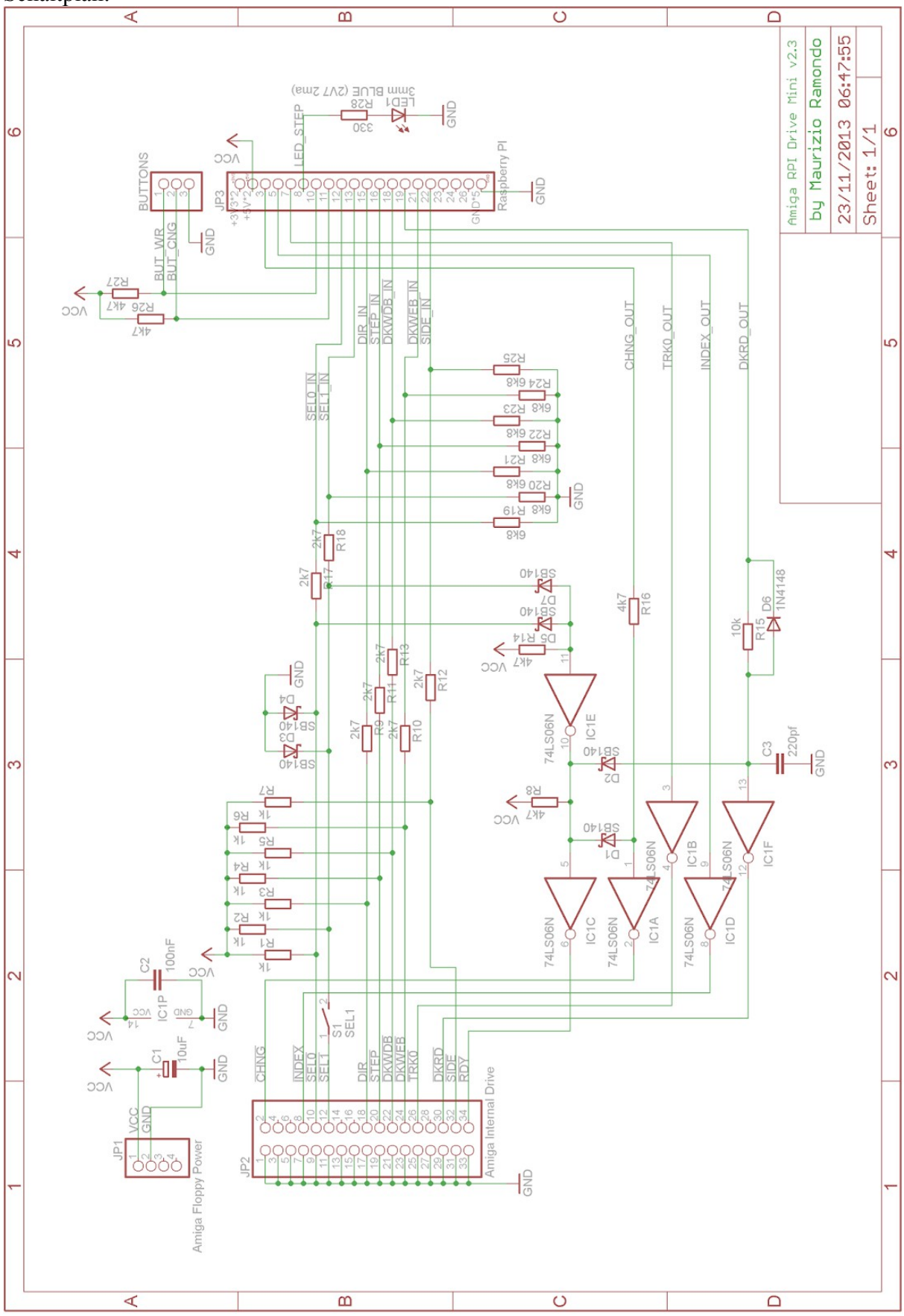
Ende 2013 hat Maurizio Ramondo einen Floppy Emulator aus einem Raspberry Pi gebaut. Dazu werden außer einem RPi (auch die A Version genügt) nur eine Zusatzschaltung mit Levelshiftern aus Widerständen und einem 74LS06 TTL IC benötigt.

An Software wird nur ein kleines Image für die SD Karte benötigt. Die Software lädt beim Start dann ein Amigaprogramm zur Belegung von bis zu 4 ADF Diskettenimages.

Bild des Emulator:



Schaltplan:


Amiga RPI Drive Mini v2.3
by Maurizio Ramondo
23/11/2013 06:47:55
Sheet: 1/1
Amiga Floppy Power
Amiga Internal Drive
Raspberry PI
BUTTONS
3mm BLUE (2V7 2ma)
74LS06N
1N4148
220pf
100nF
10uF
VCC
GND
CHNG_OUT
TRK0_OUT
INDEX_OUT
DKRD_OUT
DIR_IN
STEP_IN
DKWDB_IN
DKWEB_IN
SIDE_IN
SEL0_IN
SEL1_IN
LED_STEP
BUT_WR
BUT_CNG

Cortex Amiga Floppy Emulator

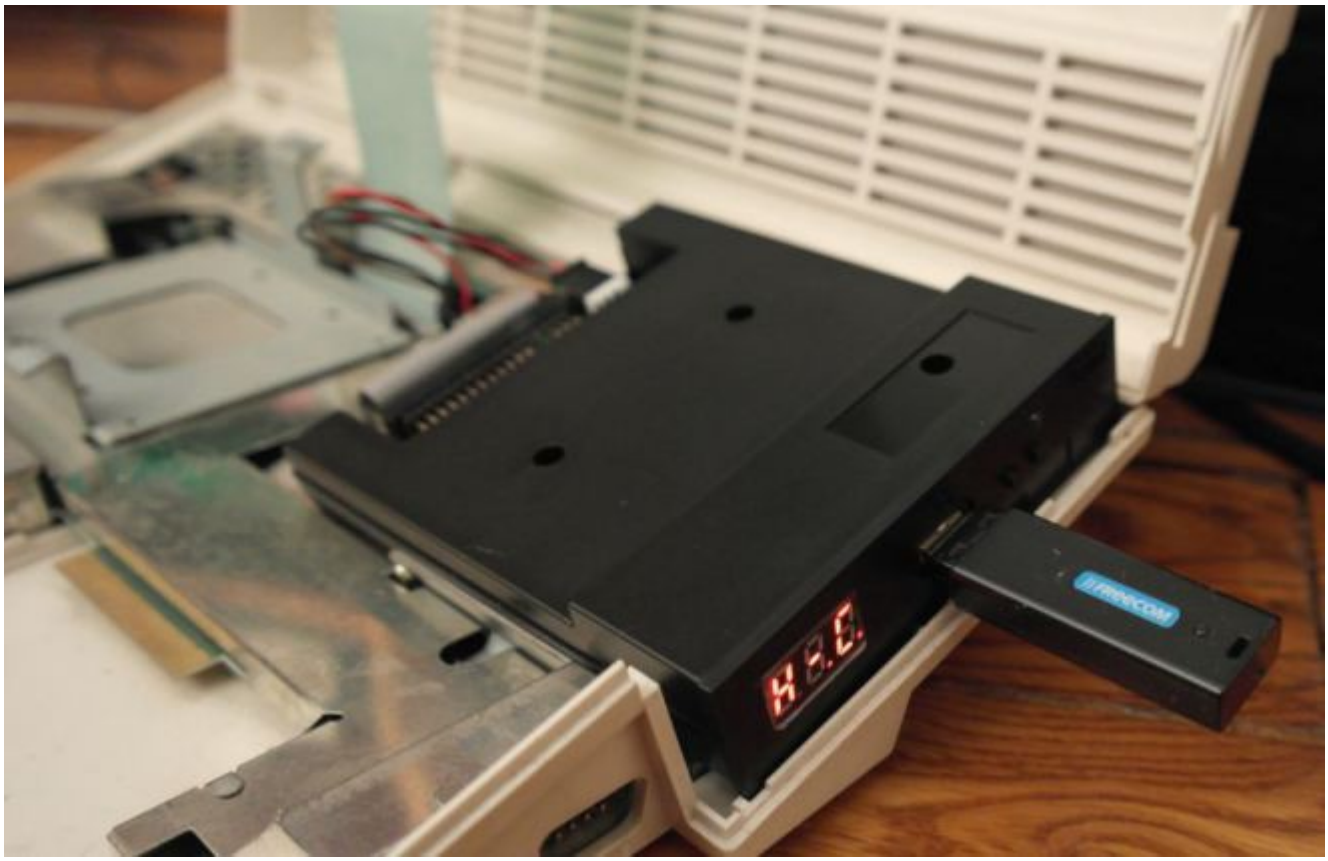
Link:
http://cortexamigafloppydrive.wordpress.com/

Herve Messinger hat den gut verfügbaren PC Floppy Emulator der Fa. Gotek durch neue Firmware zu einem Floppy Emulator für den Amiga umgebaut.

Die Gotek Floppy Emulatoren nutzen ein ARM basierenden STM32 CPU die genügend Leistung bietet.

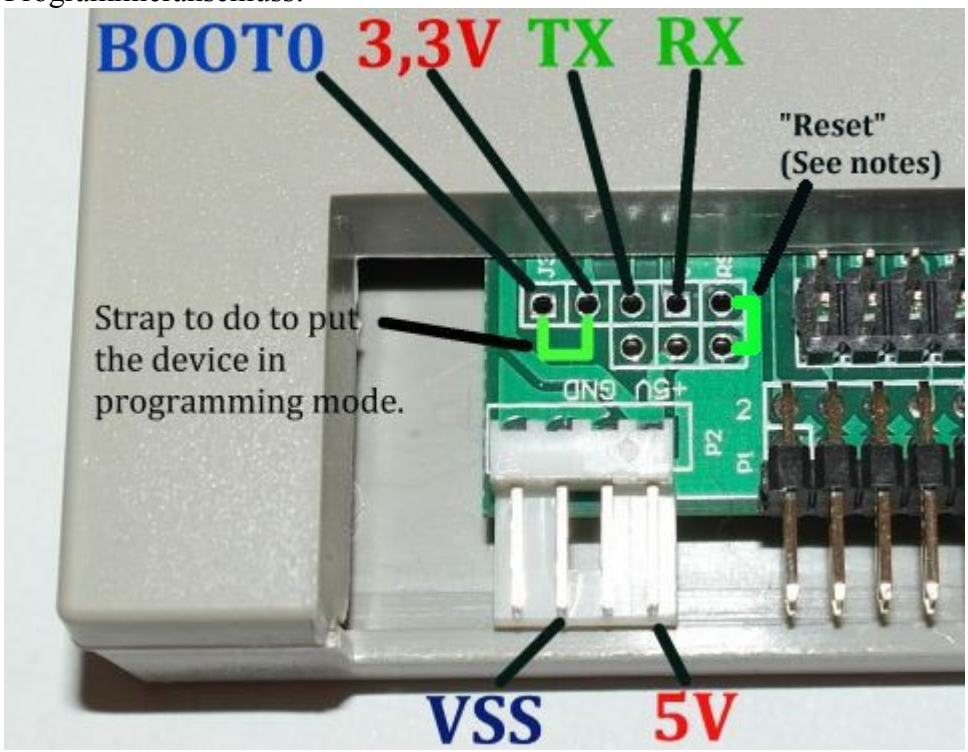
Das Gotek System gibt es bei bekannten Elektronik Händlern und Auktionsplattformen für 20-25€.

Nach Update der Firmware, werden neben der speziellen SELECTOR.ADF noch bis zu 999 weitere ADF Images auf den USB Stick (FAT32) kopiert. Nach dem Start des Amiga findet die Auswahl der ADF's statt, die sich dann über die Taster (und ggf. LED Anzeige) unter Position 001-999 mounten lassen.



Zuerst muss die original Firmware ersetzt werden. Dazu wird die neue Firmware (HEX), ein USB-TTL Dongle und das Tool STMicroelectonics Flash loader benötigt.

Programmieranschluss:



Nötige Verbindungen:

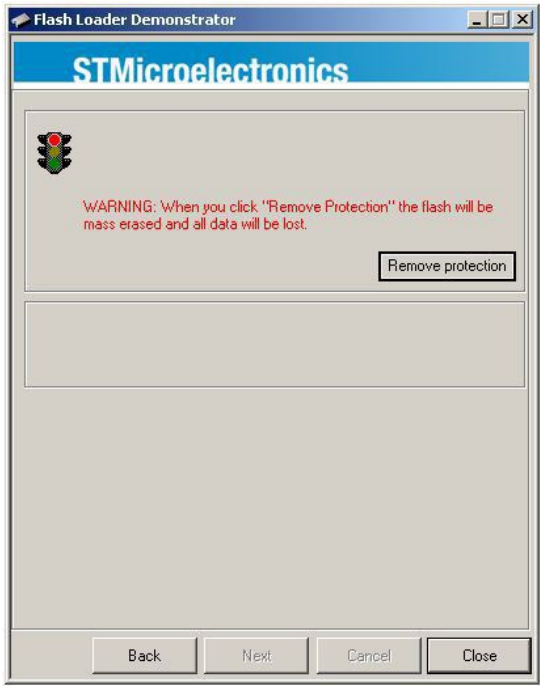
- Connect BOOT0 to 3.3V : Das bringt das Gerät in den Firmware Update Modus.
- Connect TX to the RX wire of the 3.3V USB TTL Dongle.
- Connect RX to the TX wire of the 3.3V USB TTL Dongle.
- Connect VSS to the ground wire of the USB TTL Dongle.
- Connect the 5V to a 5V source. Am USB TTL Dongle sollten +5V abgreifbar sein.

Nun die Verbindung mit dem PC herstellen und das STMicroelectonics Flash loader Tool starten.

Auswahl des COM Ports. Next.

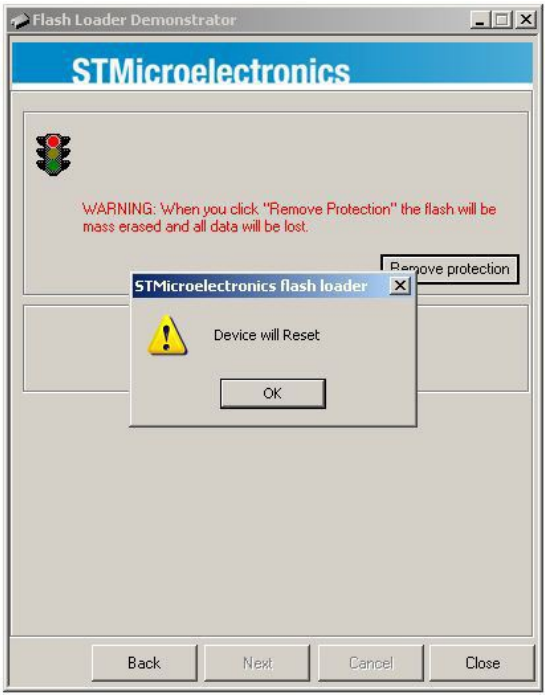
(Falls keine Verbindung zustande kommt, ggf. 1s das RST Signal mit GND verbinden)

Wenn die Verbindung steht, sollte sich folgender Bildschirm ergeben:

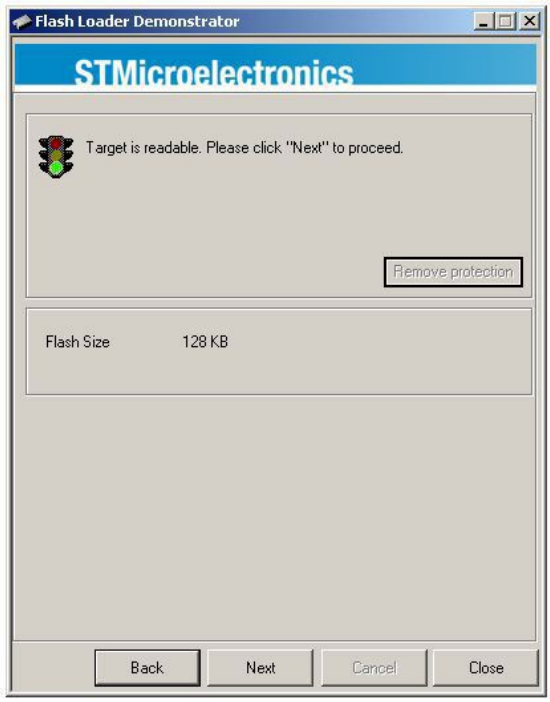


Da die Gotek Firmware gegen auslesen geschützt ist, erfolgt die Frage, ob diese Schutz entfernt werden soll, was gleichbedeutend mit dem Löschen der orig. Firmware ist!

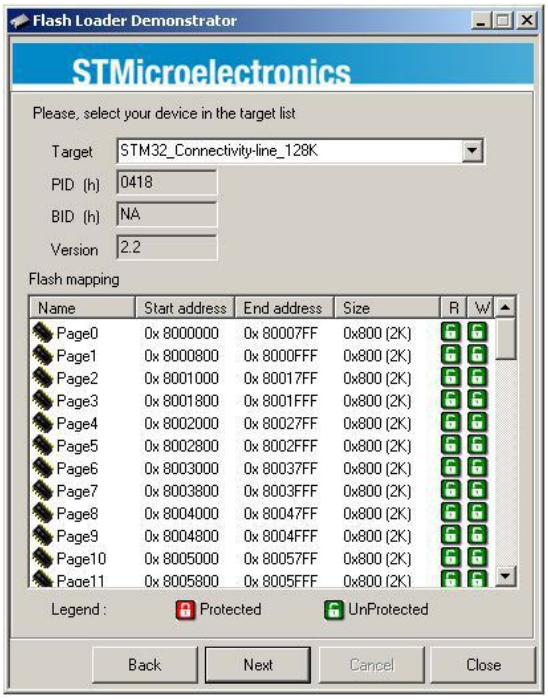
Klicke “Remove Protection”



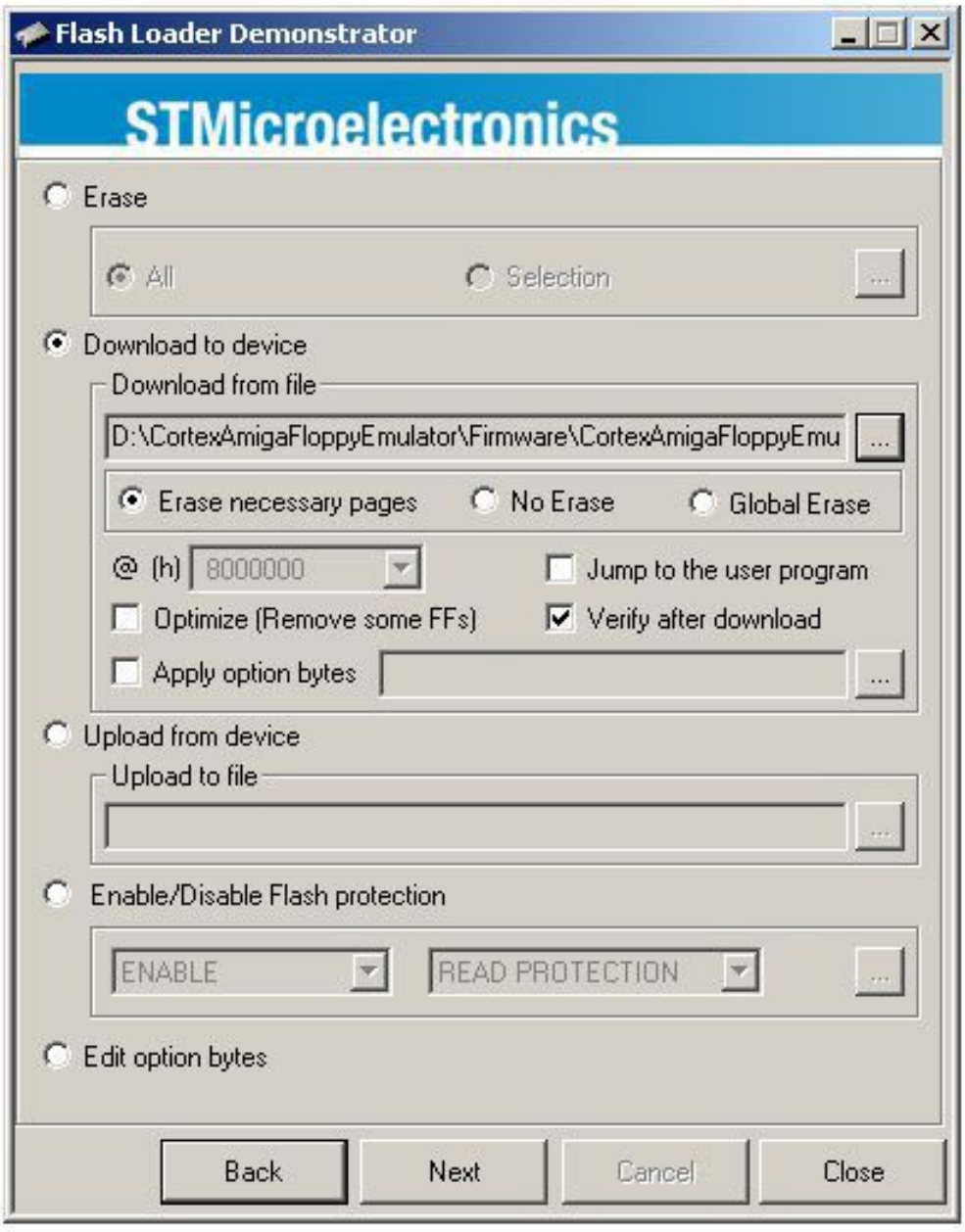
Klicke "OK"



Klicke “Next”.

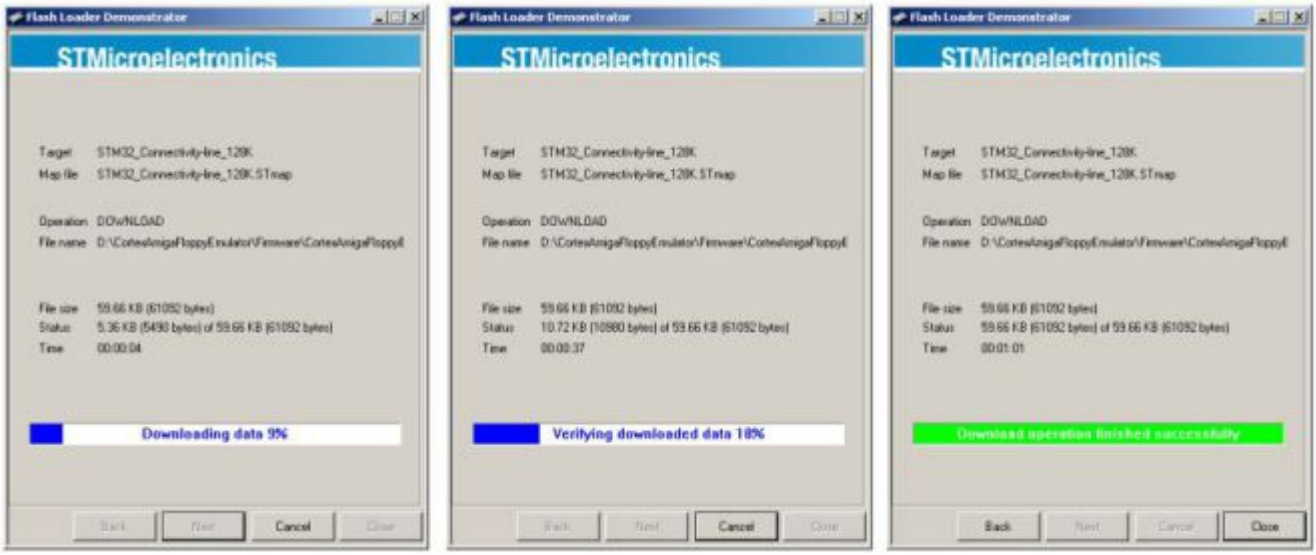

Flash Loader Demonstrator
STMicroelectronics
Please, select your device in the target list
Target
STM32_Connectivity-line_128K
PID (h)
0418
BID (h)
NA
Version
2.2
Flash mapping
Name
Start address
End address
Size
R
W
Page0 0x 8000000 0x 80007FF 0x800 (2K)
Page1 0x 8000800 0x 8000FFF 0x800 (2K)
Page2 0x 8001000 0x 80017FF 0x800 (2K)
Page3 0x 8001800 0x 8001FFF 0x800 (2K)
Page4 0x 8002000 0x 80027FF 0x800 (2K)
Page5 0x 8002800 0x 8002FFF 0x800 (2K)
Page6 0x 8003000 0x 80037FF 0x800 (2K)
Page7 0x 8003800 0x 8003FFF 0x800 (2K)
Page8 0x 8004000 0x 80047FF 0x800 (2K)
Page9 0x 8004800 0x 8004FFF 0x800 (2K)
Page10 0x 8005000 0x 80057FF 0x800 (2K)
Page11 0x 8005800 0x 8005FFF 0x800 (2K)
Legend :
Protected
UnProtected
Back
Next
Cancel
Close

Klicke “Next”.



Nun neue Firmware (HEX) wählen und Einträge analog obigem Bildschirmkopie einstellen.

Klicke “Next” -> Der Flash Update beginnt (dauert ca. 1/2 Minute).



Wenn man obige Ausgabe sieht, läuft alles gut und der neue Amiga Floppy Emulator ist bereit zum Einsatz.

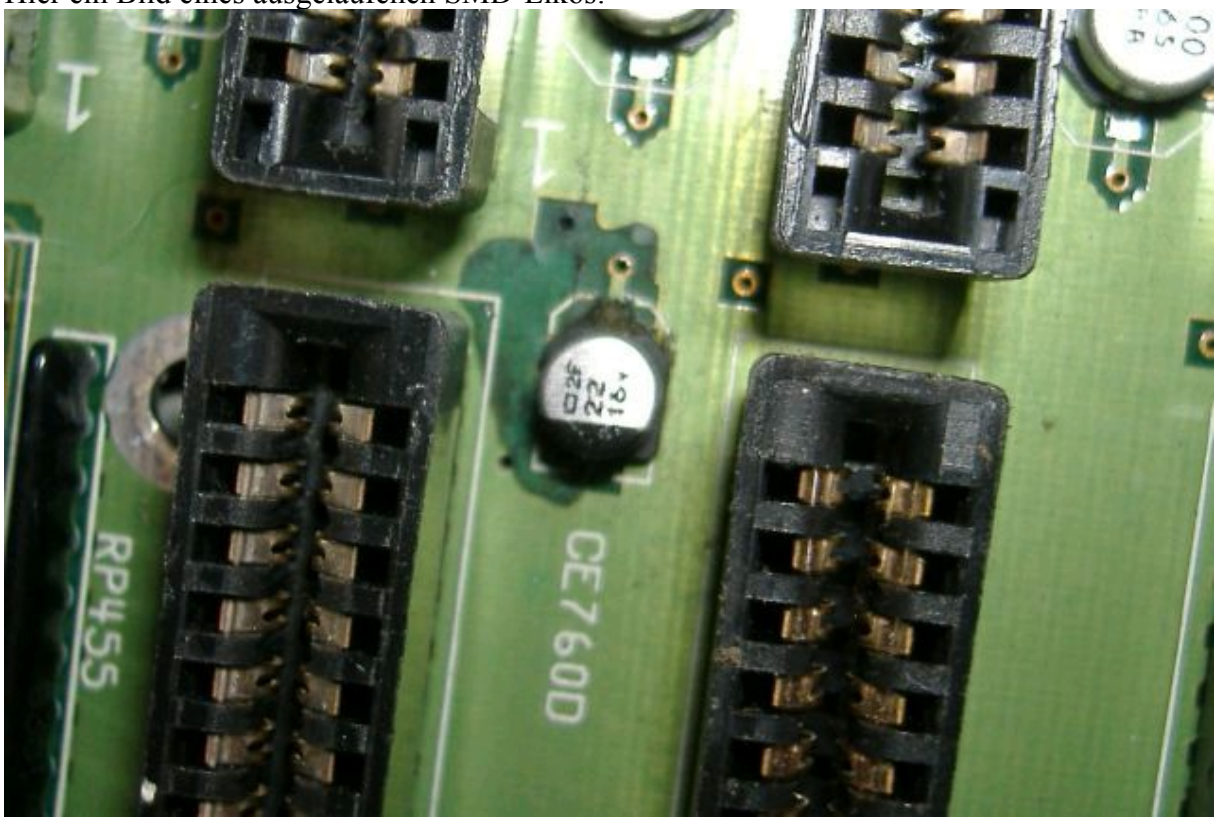
Nicht vergessen der Jumper auf die S0 Position zu stellen für DF0 Selektion.

ELKO’s an SMD Amiga’s tauschen

Von Sascha van Wahnem (AmigaGTI)

Elektrolytkondensatoren (Elko’s) altern! Dabei können sie auslaufen und durch ihren ätzenden Inhalt auch ihr Umfeld zerstören! Aus diesem Grund sollten Elko’s von Zeit zu Zeit einer optischen Kontrolle unterworfen werden und ggf. gegen frische ausgetauscht werden.

Hier ein Bild eines ausgelaufenen SMD-Elkos:



Ersatzteilliste für Elkos im Amiga 600

SMD Elkos

Bezeichnung:	Kapazität:	Spannung:	Bestellnummer Reichelt:	Bestellnummer Kessler:
C821	47 uF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C822	47 uF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C235	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C236	100 uF	16 v	SMD ELKO 100/16	ES 100/16-SMD
C239	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C214	10 uF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C460	10 uF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C???	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C459	22 uF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C324	22 uF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C306	10 uF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C612	10 uF	25 V	SMD ELKO 10/25	ES 10/25-SMD
C304	22 uF	35 V	SMD ELKO 22/35	ES 22/35-SMD
C303	22 uF	35 V	SMD ELKO 22/35	ES 22/35-SMD

Becher Elkos

Bezeichnung:	Kapazität:	Spannung:	Bestellnummer Reichelt:	Bestellnummer Kessler:
C307	470 uF	16 V	RAD 470/16	EST 470/16
C237	470 uF	16 V	RAD 470/16	EST 470/16
C408	1000uF	10 v	RAD 1000/10	EST 1000/10
C811	1000 uF	10 V	RAD 1000/10	EST 1000/10

Ersatzteilliste für Elkos im Amiga 1200

SMD Elkos

Bezeichnung:	Kapazität:	Spannung:	Bestellnummer Reichelt:	Bestellnummer Kessler:
C822	47 uF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C306	10 uF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C303	22 uF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C304	22 uF	25 v	SMD ELKO 22/25	ES 22/25-SMD
C???	22 uF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C324	22 uF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C??7	10 uF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C821	47 uF	16 V	SMD ELKO 47/16	ES 47/16-SMD
C235	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C459	22 uF	25 V	SMD ELKO 22/25	ES 22/25-SMD
C214	10 uF	35 V	SMD ELKO 10/35	ES 10/35-SMD
C239	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C236	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD
C409	100 uF	16 V	SMD ELKO 100/16	ES 100/16-SMD

Becher Elkos

Bezeichnung:	Kapazität:	Spannung:	Bestellnummer Reichelt:	Bestellnummer Kessler:
C811	1000 uF	10 V	RAD 1000/10	EST 1000/10
C237	470 uF	16 V	RAD 470/16	EST 470/16
C408	1000uF	10 v	RAD 1000/10	EST 1000/10
C307	470 uF	16 V	RAD 470/16	EST 470/16

Das gilt analog auch für alle anderen Elko's/SMD Elko's in anderen Systemen. Z.B auch im Amiga 4000 etc.

AT/X Netzteil für A1000

AT Netzteilsteckerbelegung:

Pin	Farbe	Bedeutung
1	Orange	Power ok +5V
2,10,11,12	Rot	+5V
3	Gelb	+12V
4	Blau	-12V
5,6,7,8	Schwarz	GND/Masse
9	Weiß	-5V

ATX Netzteilsteckerbelegung:

Pin	Farbe	Bedeutung
1,2,11	Orange	+3,3V
3,5,7,13, 15,16,17	Schwarz	GND/Masse
4,6,19,20	Rot	+5V
8	Grau	Power ok
9	Violett	+5V für Standby
10	Gelb	+12V
12	Blau	-12V
14	Grün	PS ON (gegen GND=Netzteil EIN)
18	Weiß	-5V

Achtung: Sich nicht auf diese Standardfarben verlassen! Aufschriften auf dem Netzteil kontrollieren und immer auch selbst nachmessen!
Man muß Pin 14 (PS ON) mit einem Masse-Pin verbinden, um das ATX-Netzteil einzuschalten. Ich habe ein ATX-Netzteil mit Ein-/Ausschalter verwendet und PS ON dauerhaft auf GND gelegt. ATX Netzteile benötigen eine Grundlast (also A1000/PC Mainboard oder CD-Laufwerk).

A1000 Power Connector J14:
Von Vorne gesehen. Pin 7/Grau liegt dem Modulator am nächsten)

7	Grau	TICK Signal (50Hz)
6	Rot	+5V
5	Rot	+5V
4	Schwarz	GND/Masse
3	Schwarz	GND/Masse
2	Orange	+12V
1	Violett	-5V

Benötigt vom PC Netzteil werden also +5V, +12V, -5V und GND.

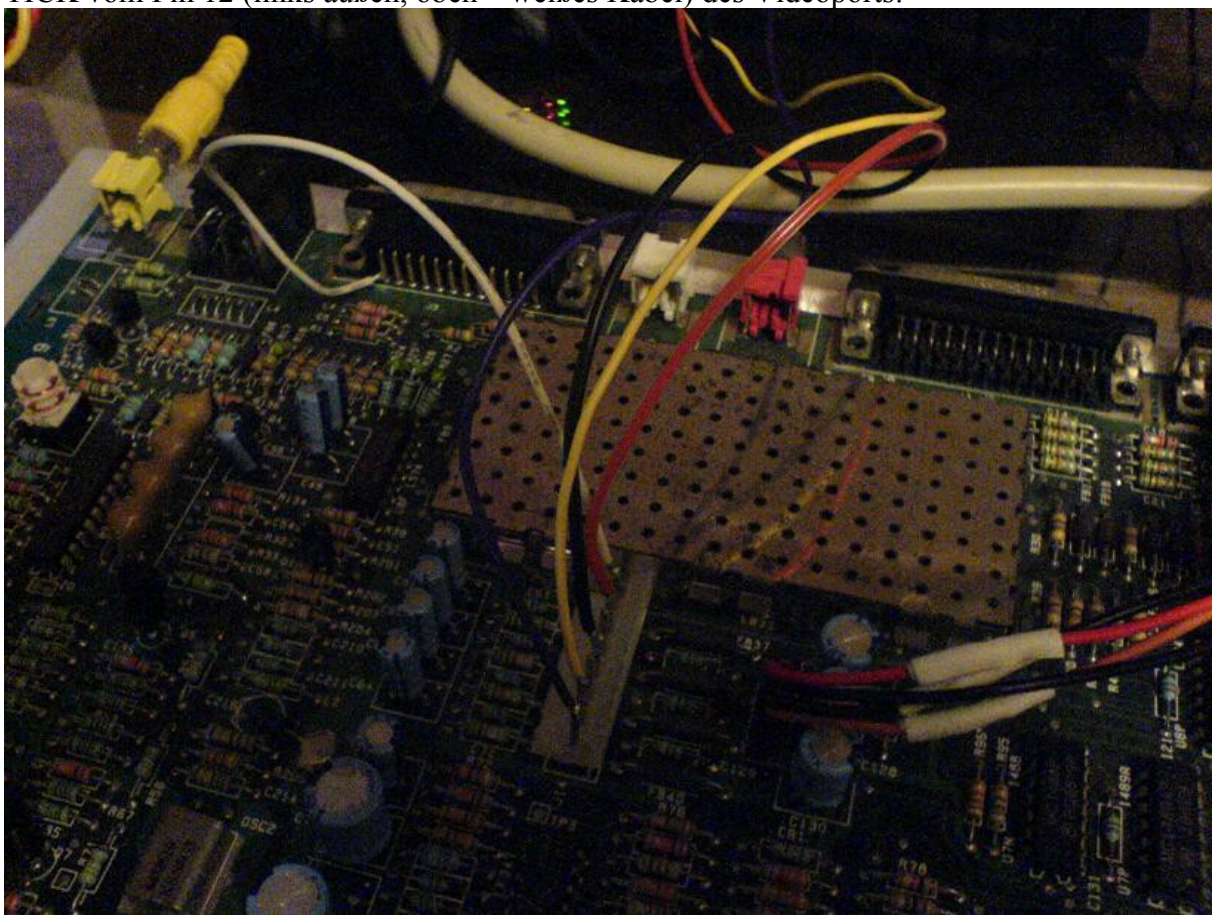
TICK Signal

Der A1000/2000 benötigt zwingend ein TICK Signal um stabil arbeiten zu können!
Dieses ist ein 50Hz (USA 60Hz) Rechecksignal.

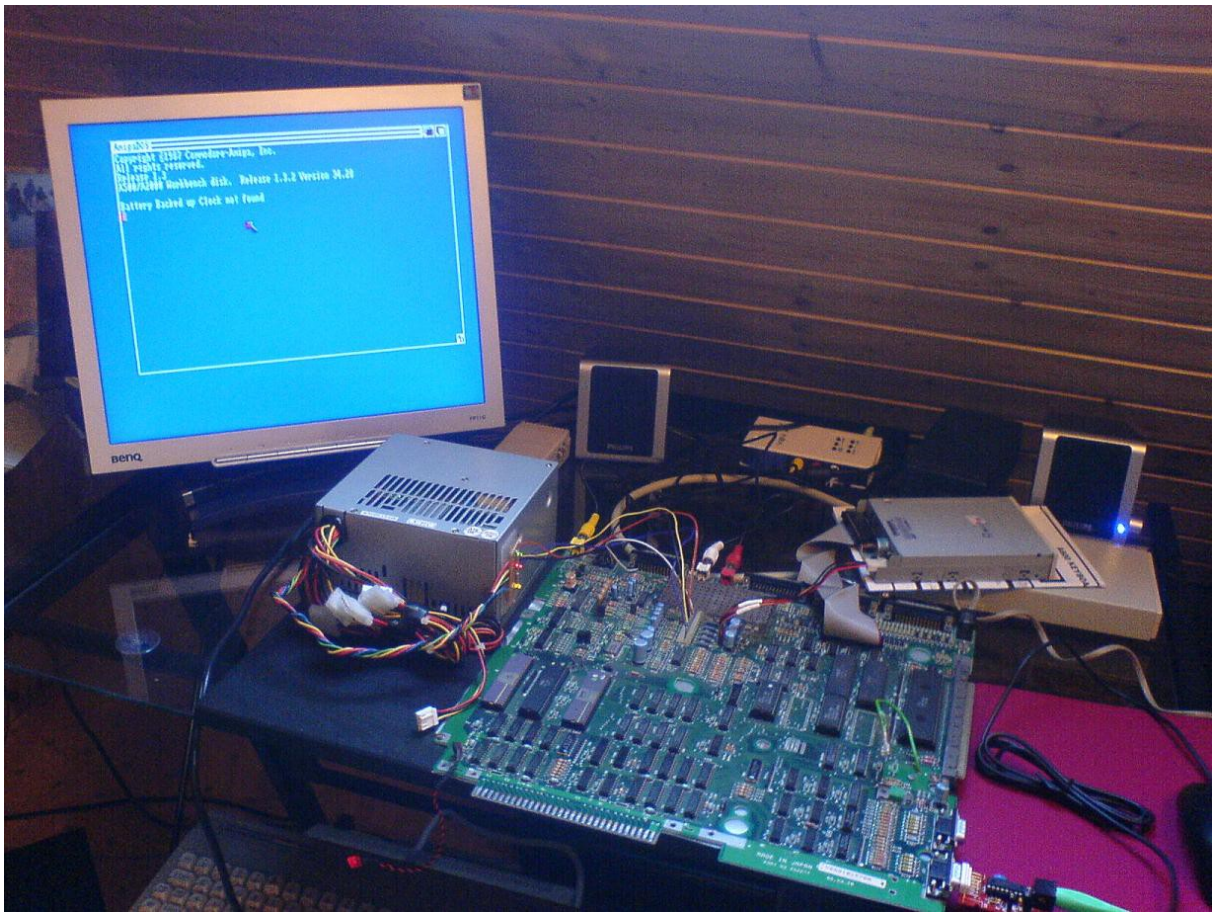
Ein solches Signal liegt an Vsync an. Dieses ist am Pin 12 vom Amiga Videoport oder an Pin 38 vom 8367 AGNUS Chip verfügbar. Dieses einfach mit dem TICK-Pin des Power Anschlusses verbinden.

Bei keiner weiteren Belastung des Signales ist das so ausreichend. Beim A500 ist das auch gemacht.

TICK vom Pin 12 (links außen, oben – weißes Kabel) des Videoports:



A1000 Brett-Edition:



Um das externe Floppy als Bootfloppy (DF0:) nutzen zu können, ‚muß ein 40-pol Zwischensockel für CIA U6S angefertigt werden, wo SEL0 – Pin 13 und SEL1 – Pin 14 Vertauscht werden (Kabel über Kreuz anschließen).

Wer möchte kann dazu natürlich auch einen Umschalter entsprechend anschließen.

Für A500 ist das U8, für A2000A ist es U11 und für den A2000B ist es U301.

Amiga am TFT anschließen

Der Anschluß eines A500 über einen A520 TV Modulator, oder eines A600 über Video-Out und Scart-Adapter an einen Fernseher dürfte klar sein. Aber evtl. möchte man seinen Amiga auch an einem TFT Monitor betreiben. Dazu sind eigendlich teure sog. Scandoubler und Flickerfixer erforderlich. Mit passendem TFT-Monitor, der auch schon bei ca. 50Hz synchronisiert und einem Video zu VGA Konverter kann man dies aber unter Umständen und mit geringere Bildqualität auch erheblich preiswerter erreichen.

Ich hatte mir dazu vor einigen Tagen den Video2VGA Konverter von Tinxi bestellt. 29,99 +VS = 36€. Nach den bisherigen Erfahrungen ich von einer mittelmäßigen, für Spiele hoffentlich noch brauchbaren Bilddarstellung ausgegangen.. aber 30€ sind ein guter Preis. Bestellen kann man dieses Gerät übrigens direkt unter www.tinxi.com - Artikel: 33003

Nach dem Anschluß über Video-In muß sagen, ich bin doch positiv überrascht. Textverabeitung sollte man damit nicht machen.. aber für Spiele durchaus gleichwertige Darstellung im Vergleich zum älteren Fernseher. Durch den VGA-In kann auch z.B. mein Minimig DE1 oder jede andere VGA Quelle parallel angeschlossen bleiben. Über einen 3,5mm Stereo-AudioY-Stecker lassen sich auch Audio vom Amiga und Minimig/PC parallel oder umzustöpseln betreiben.

Bild des Video2VGA Konverters:

Bild eines Amiga 600 am TFT:

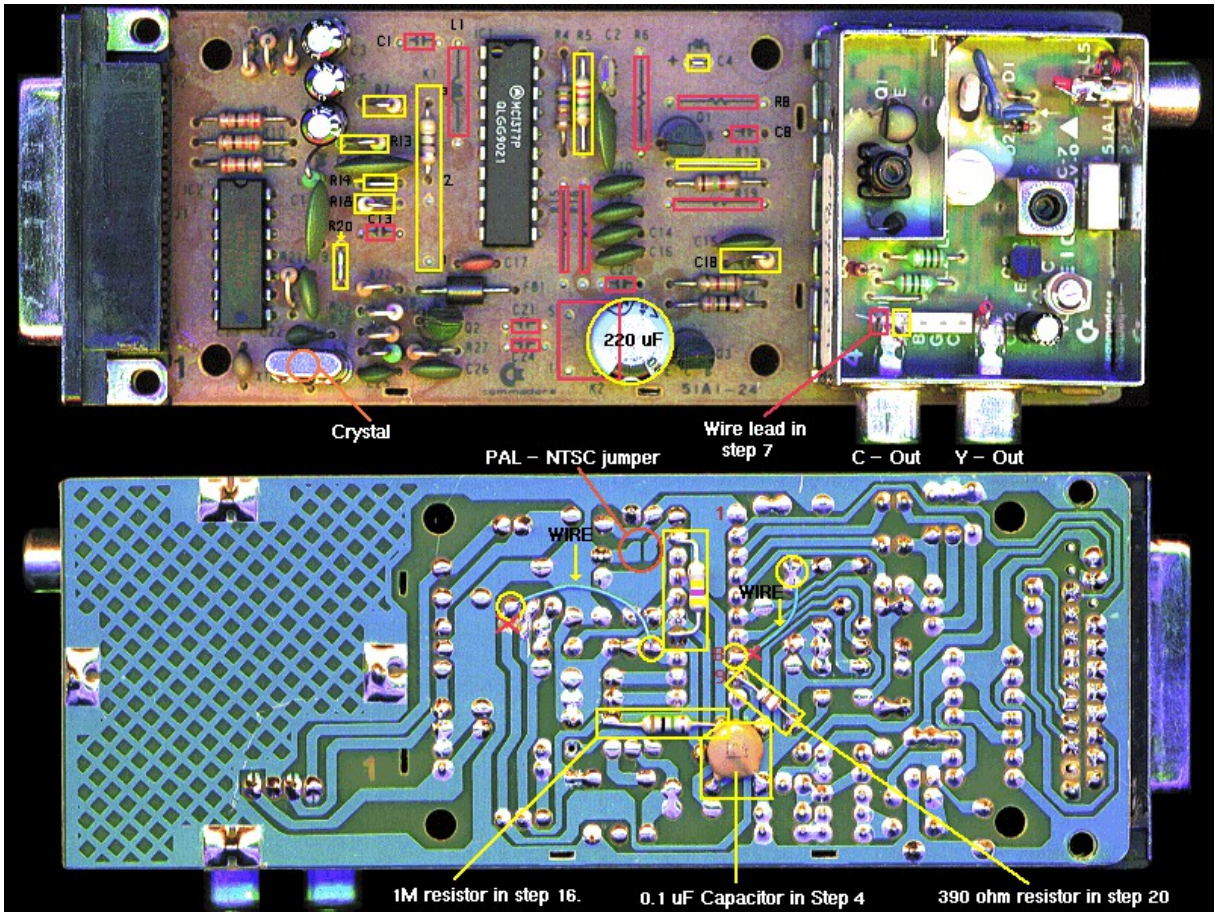
C64 und DTV gehen leider so einfach nicht, da deren Signalqualität miserabel ist.
Das Bild scrollt nach links weg und ist sehr unscharf.
Folgender Tipp kann für ein DTV für Abhilfe sorgen:
POKE 53311,1:POKE 53326,2 setzt $D04E/"Scan line timing adjust" auf 2
(0 ist der Standard).

A520 TV-Modulator auf S-Video umbauen

David Myers hat eine Umbauanleitung entwickelt, um einen A520 TV-Modulator so umzubauen, das man ein S-Video Signal erhält, das eine bessere Signalqualität als das FBAS Signal hat.

Achtung: Als TV-Modulator ist der A520 nach dem Umbau nicht mehr zu gebrauchen!

Das Bild wurde von David freundlicherweise zur Verfügung gestellt.



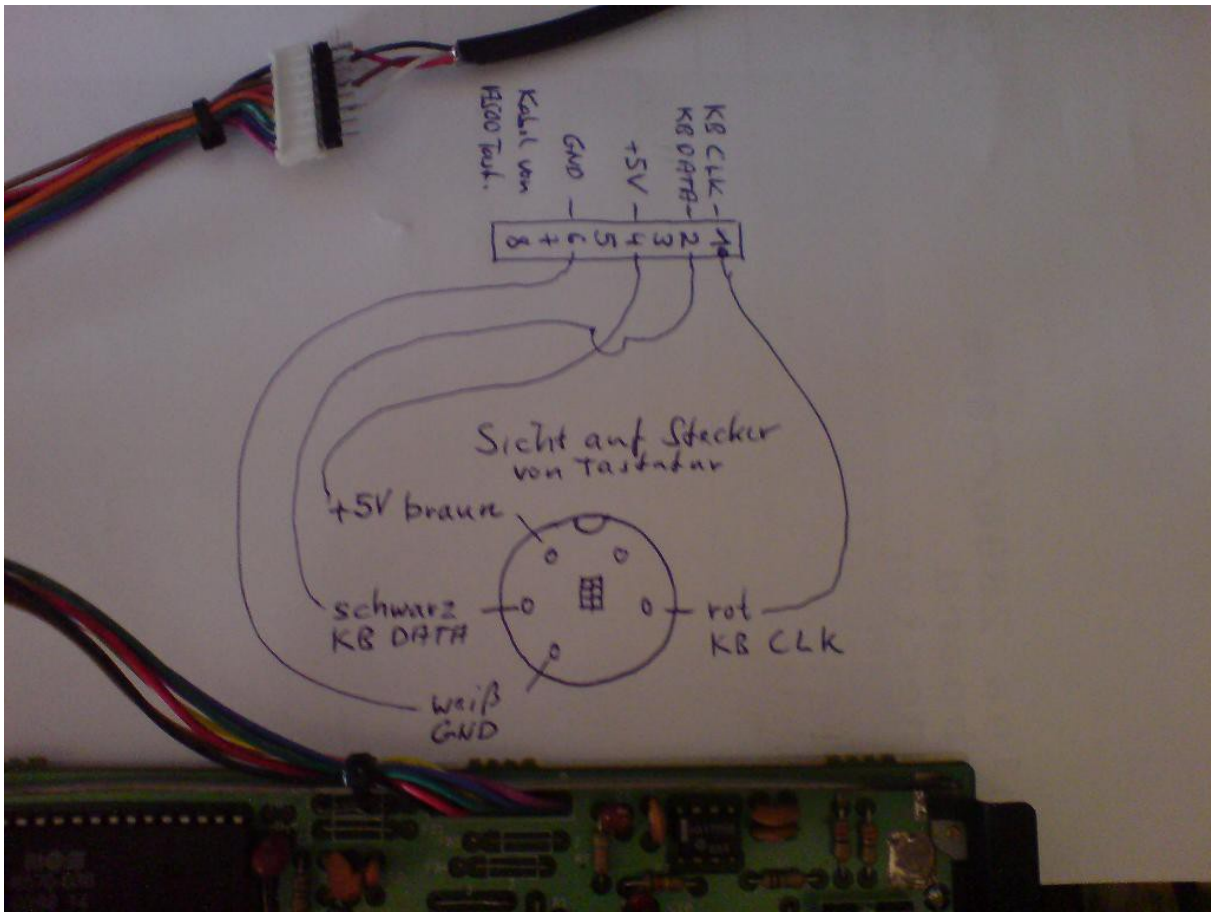
Hier die übersetzte Schritt für Schritt Anleitung.

- C1 und L1 entfernen (ist evtl. gar nicht bestückt).
- Die Spule K2 und C8, C13, C20, C21, C24, R6, R8, R15, R16, R19 entfernen.
- K1 (delay line) entfernen und einen 470 Ohm Widerstand zw. Pin 2 + 3 einsetzen.
- Einen 0.1 uF Kondensator zw. Pin 10 vom IC und Masse anschließen (Unter der Platine).
- C18 entfernen und durch einen 330 Ohm Widerstand ersetzen.
- Leiterbahn zw. Basis (B) von Transistor Q1 (siehe rotes Kreuz auf Bild) durchtrennen.
- Pin 13 vom IC mit der nun isolierten Basis (B) von Q1 verbinden (Unter der Platine).
- Die Verbindung des Audio-Input Cinch Stecker entfernen (Im Modulator – markiert mit dem roten Quadrat im Bild).
- Nun den Cinch-Stecker (Audio-Input) mit einem 100-180 Ohm Widerstand (*) zum Punkt 'V' der weißen Anschlußreihe verbinden (siehe gelbes Quadrat).
- R14 durch eine Drahtbrücke ersetzen.
- Leiterbahn zu Pin 8 vom IC durchtrennen (markiert mit Kreuz Unterseite Platine).
- Pin 8 des IC nun mit Pin 3 der ehemaligen K1 (delay line) mit einem Kabel verbinden (Pin 2+3 sind jetzt mit einem 470 Ohm Widerstand überbrückt – siehe Schritt 3.).
- R7 durch einen 180 Ohm Widerstand ersetzen.
- C4, R20, R13 entfernen und durch eine Drahtbrücke ersetzen.
- R11 durch einen 1KOhm Widerstand ersetzen.
- Einen 220 Ohm Widerstand an der leeren R18 Position einsetzen.
- Einen 1MOhm Widerstand von Pin 17 des IC zu +12 Volt legen (Unterseite Platine).
- R5 entfernen (falls bestückt) und durch einen 2.2MOhm Widerstand ersetzen.
- Einen 470KOhm Widerstand von Pin 14 zu Pin 19 des IC löten (Unterseite Platine)..
- Einen 220uF Kondensator zw. Pin 3 + 4 der ehemaligen Spule K2 einsetzen. +Pol an 4.
- Einen 390 Ohm Widerstand von Pin 9 des IC zu Pin 1 der ehemaligen K1 (delay line) legen (Unterseite Platine).

Dann ein entsprechendes S-Videokabel auf 2xCinch (Y/C) anfertigen. Fertig.
(*) In der original Anleitung ist hier nur eine Drahtverbindung vorgesehen. Dies kann aber unter bestimmten Umständen, den dadurch ungeschützten Transsistor Q1 zerstören (Ersatztyp z.B TIS90).

Amiga 500 Tastatur am CDTV

Als Stecker habe ich ein Din-6 Stecker+Kabel von einer defekten PS/2 Tastatur genommen. Einen Pin (unten rechts - Draufsicht Stecker) abgekniffen und auch der Mittel-Plastikpin ist zu dick, muß also auch entfernt werden. In die Buchse am A500 Tastaturkabel dann eine 8-pol Lötpfostenreihe eingesteckt und über 4 Kabel mit dem umgebauten PS/2 Stecker verbunden (+5V, GND, KB CLK und KB Data).



Verkaufszahlen Commodore Deutschland

Herausgegeben von der Marketing Abteilung Commodore Deutschland:

System	Stückzahl (DE)
Amiga CD32	25000
Amiga CDTV	25800
Amiga 500	1081000
Amiga 500+	79500
Amiga 600	193000
Amiga 1000	27500
Amiga 1200	95500
Amiga 2000	124500
Amiga 3000	8300
Amiga 4000	11300
C 128	284300
C16/116	286500
C64	3050000

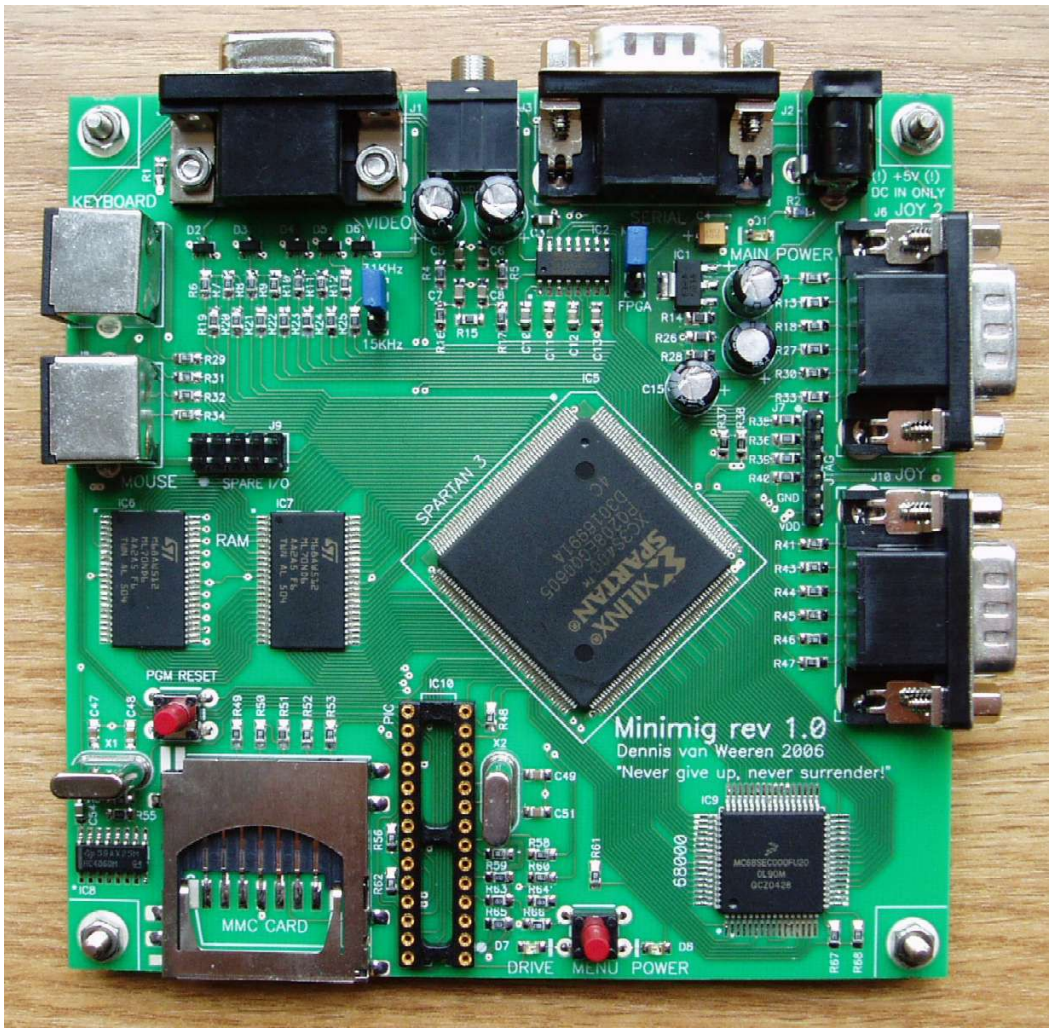
Daraus ergeben sich die relative Seltenheit von z.B A3000/4000 (dafür muß man heute sicher über 200-300€ zahlen) und die doch noch gute und relativ preiswerte Verfügbarkeit der A500 Systeme auf dem Gebrauchtmarkt.

Minimig

Minimig (Kurzform für Mini Amiga) ist eine Open Source Reimplementation eines Amiga 500 in FPGA (Field Programmable Gate Array).

Minimig wurde entworfen und gebaut vom dem Niederländischen Ingenieur Dennis van Weeren. Auslöser waren nicht endende Diskussionen in der Amiga Gemeinschaft, ob sowas möglich wäre. Die Entwicklungszeit ging von ca. Januar 2005 bis ca. Mitte 2007. Am 25.07.2007 wurden die Projektdaten (Sourcecode, Schaltplan) unter der GNU General Public License Version 3 gestellt.

Seitdem sind einige Minimig's in Eigenregie gebaut worden. Aber auch Acube Systems aus Italien läßt Minimig's bauen und vertreibt diese. In Deutschland werden diese durch Vesalia vertrieben.



Minimig auf Altera DE1 Board

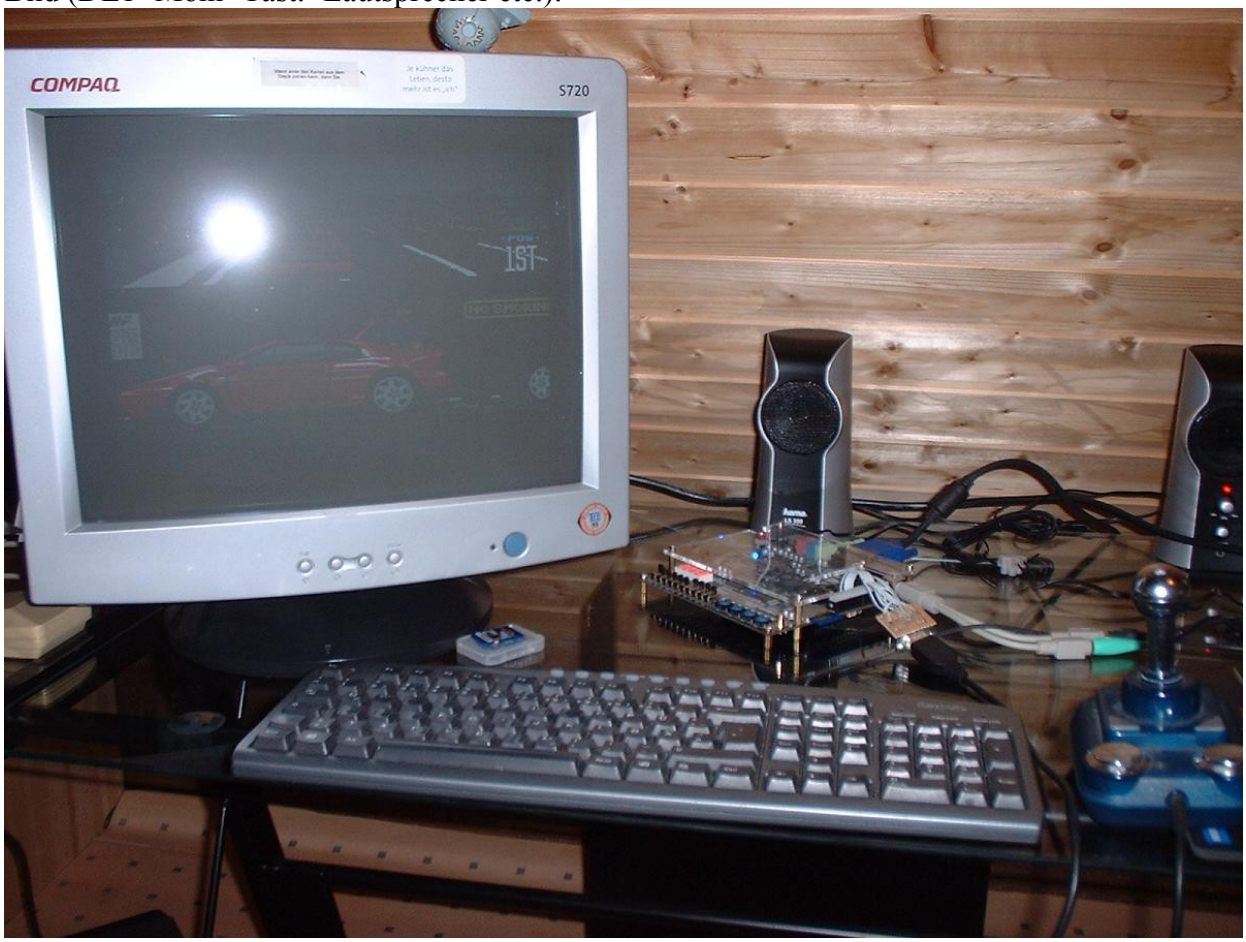
Minimig von Dennis van Weeren ist die Reimplementation eines Amiga 500 in FPGA. Von tobiflex (Tobias Gubener) mit einem 68000 Core erweitert und deshalb lauffähig auf DE1/2 Boards von Altera. Es wird noch eine Kickstart-ROM Datei mit Namen 'kick.rom' und Größe 512kb im Rootverzeichnis benötigt (Kick 1.3 = 256k; einfach mit copy /b kick13.rom+kick13.rom kick.rom zu einer 512kb Datei zusammen kopieren.) Im core Archive ist die Datei 'spihost.rom' enthalten. Diese muß ebenfalls auf die SD Karte ins Rootverzeichnis. Dann noch ein paar ADF Dateien drauf und schon kanns losgehen..

Die Schiebeschalter SW 0-3 und SW9 müßen oben stehen.

SW[9]	Monitorfrequenzumschaltung.
SW[0]	aus-an löst einen Komplett-Reset aus.
SW[3..0]	müssen an sein - damit lassen sich einzelne Teile des Cores resetten.
SW[8..4]	ungenutzt.

Debugausgaben kommen über die RS232 mit 115000 Baud und 8N1.

Bild (DE1+Moni+Tast.+Lautsprecher etc.):

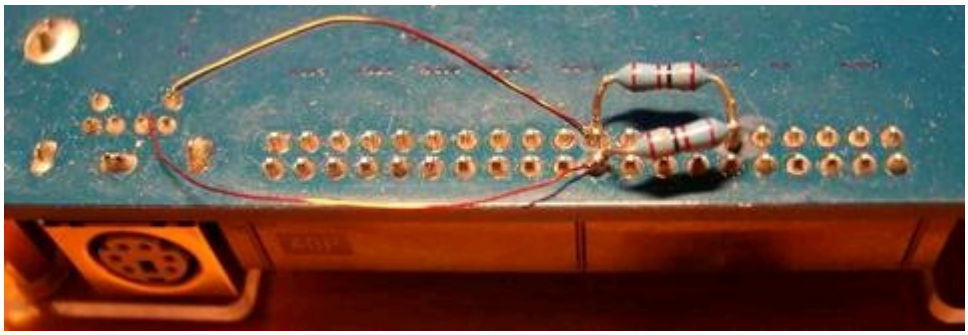


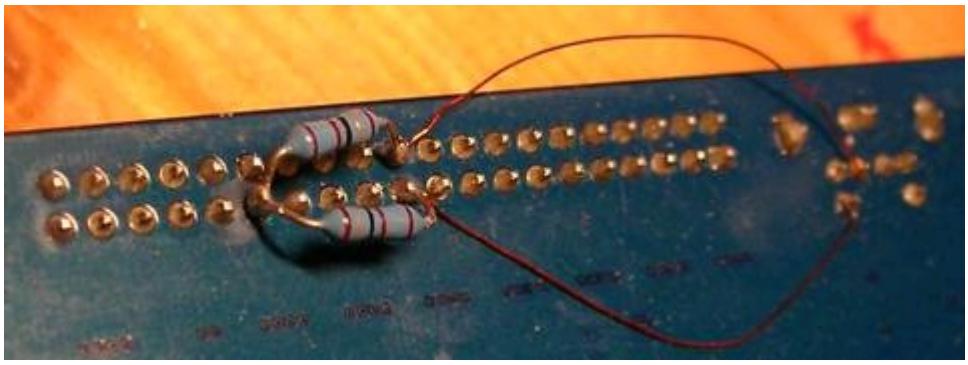
Die SOF Datei ist zum einmaligen Konfigurieren des FPGA, nach dem Ausschalten ist sie wieder weg. Die Programmiersoftware muß dazu im Mode JTAG stehen und ein Häkchen bei PROGRAM/Configure beim geladenen SOF File.

Um die POF Datei ins Start-Eeprom zu programmieren muß der Schiebeschalter links oberhalb der LEDs Richtung LEDs geschoben werden. Der Programmiermode muß dazu auf Active serial Programmierung stehen. Der Vorgang dauert um einiges länger als die normale Konfiguaration. Haken bei PROGRAM/Configure nicht vergessen. Danach den Schiebeschalter wieder Richtung USB Stecker.

Zum Laden der Pof/Sof Datei benötigt man nicht die volle Quartus II Web Edition (800MB), sondern es gibt auch einen Programmer Only mit ca. 60MB. Für letzteren benötigt man auch keine spezielle Lizenz, welche bei der Web Edition beantragt werden muß (kostenfrei – aber zeitlich begrenzte Gültigkeit).

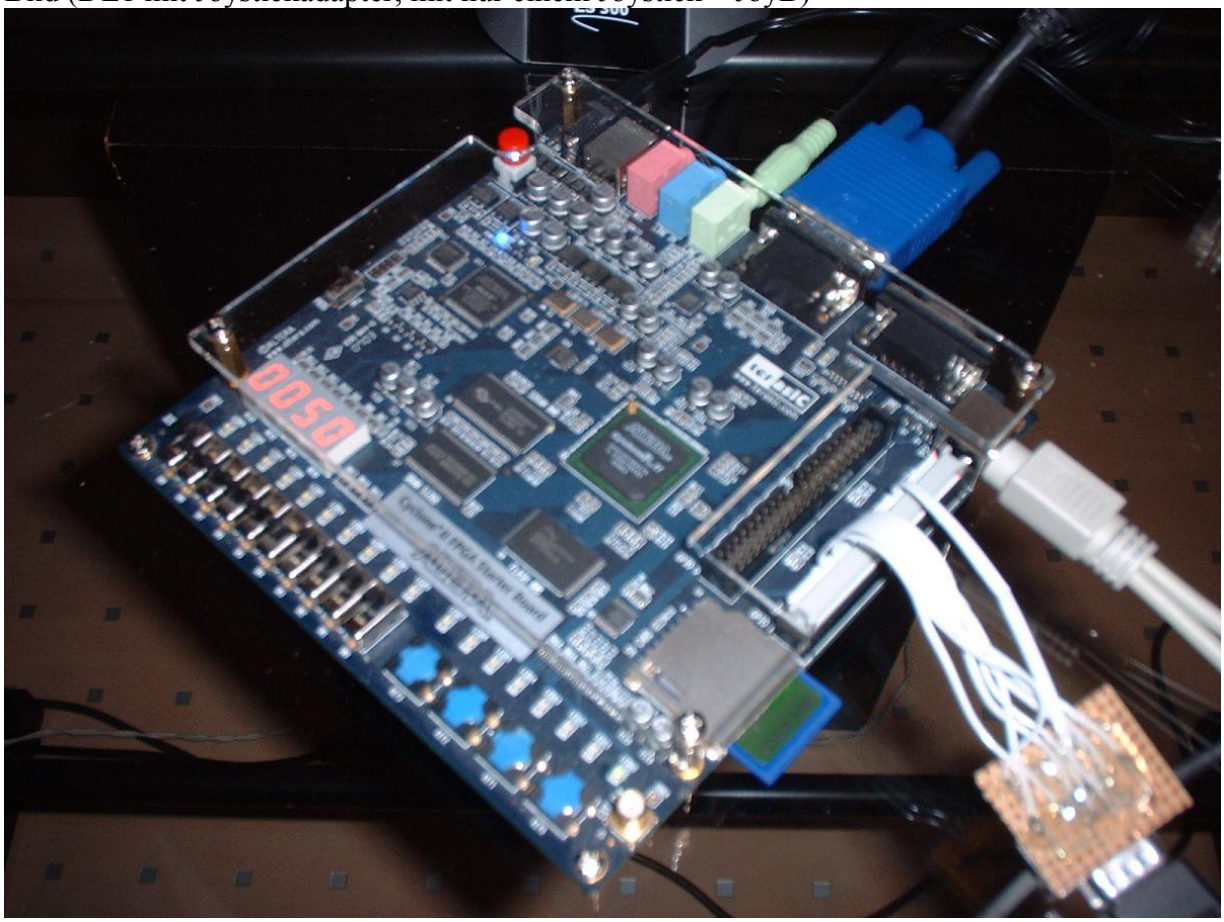
PS/2 Mauserweiterung an vorhandener PS/2 Tastaturbuchse (für Y-Kabel – Achtung: Anschluß umgekeht zur Beschriftung! Tastatur an Mausbuchse; Maus an Tastaturbuchse):



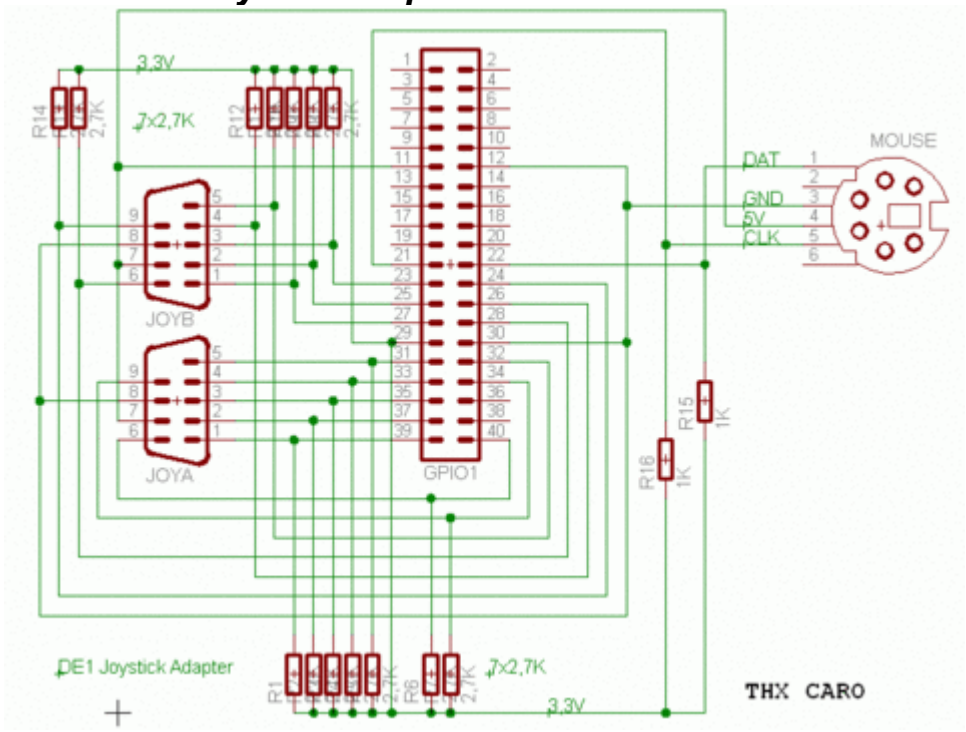


(2 x 1k Widerstand)

Bild (DE1 mit Joystickadapter, mit nur einem Joystick – JoyB)



Schaltbild Joystickadapter



Tastaturbelegung PS/2

F12 - Minimig-Menü aufrufen/beenden
PgUp - Menue hoch
PgDn - Menue runter
Pos1 - Auswahl

Micro KBD - Micro-Tastaturersatz für Minimig

Man nehme: Atmega8, 5 Taster, PS/2-Kabel+Stecker, Lochraster+Kabel, micro_kbd.hex und fertig ist die Miniatur Tastaturersatz fürs Minimig...

C0 = F12
C1 = PgUp
C2 = PgDn
C3 = Home/Pos1
C4 = F1

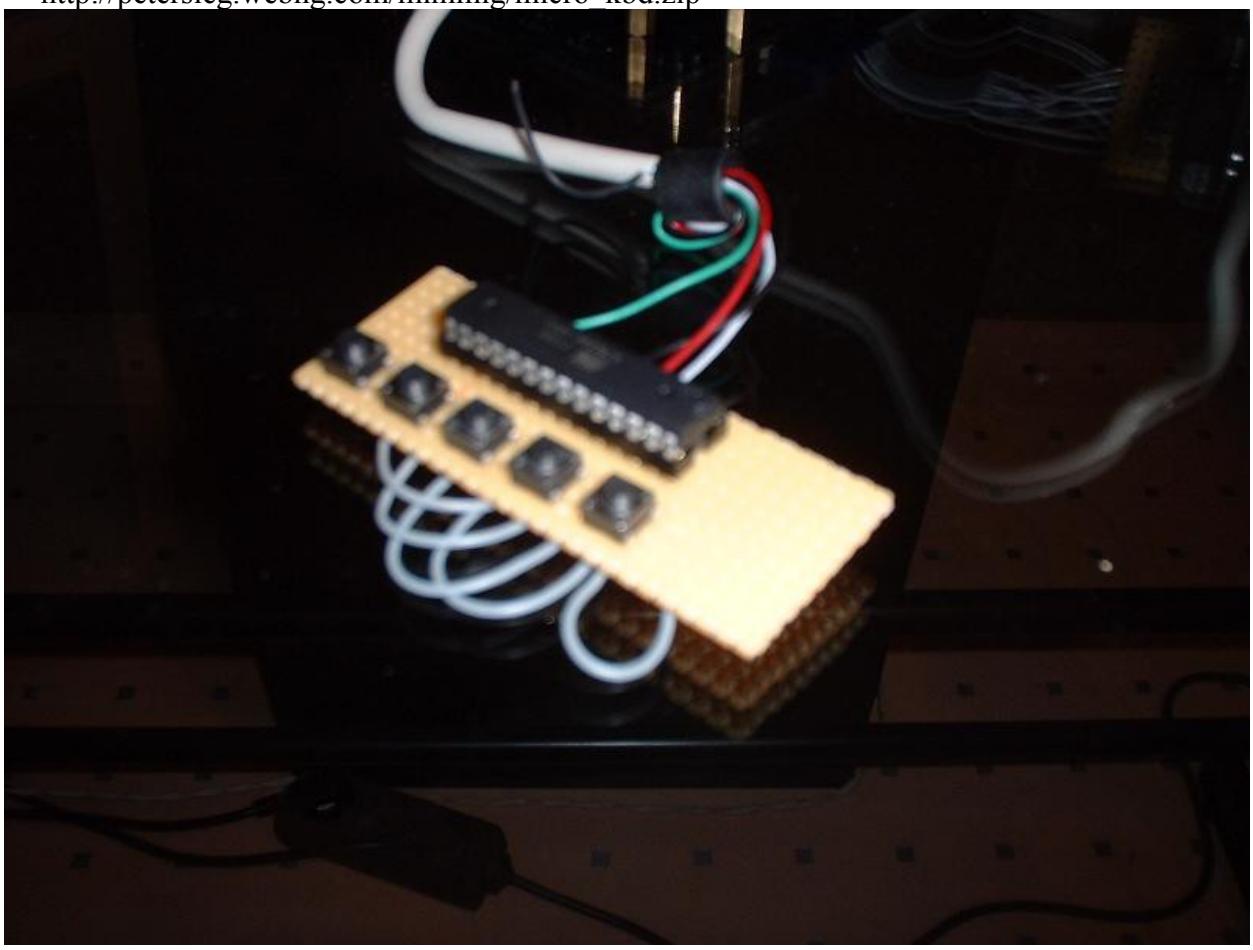
B0 = Clock-PS/2
D3 = Data-PS/2

GND und +5V zw. PS/2 und AVR verbinden. Cn = PortC Taster gegen GND.

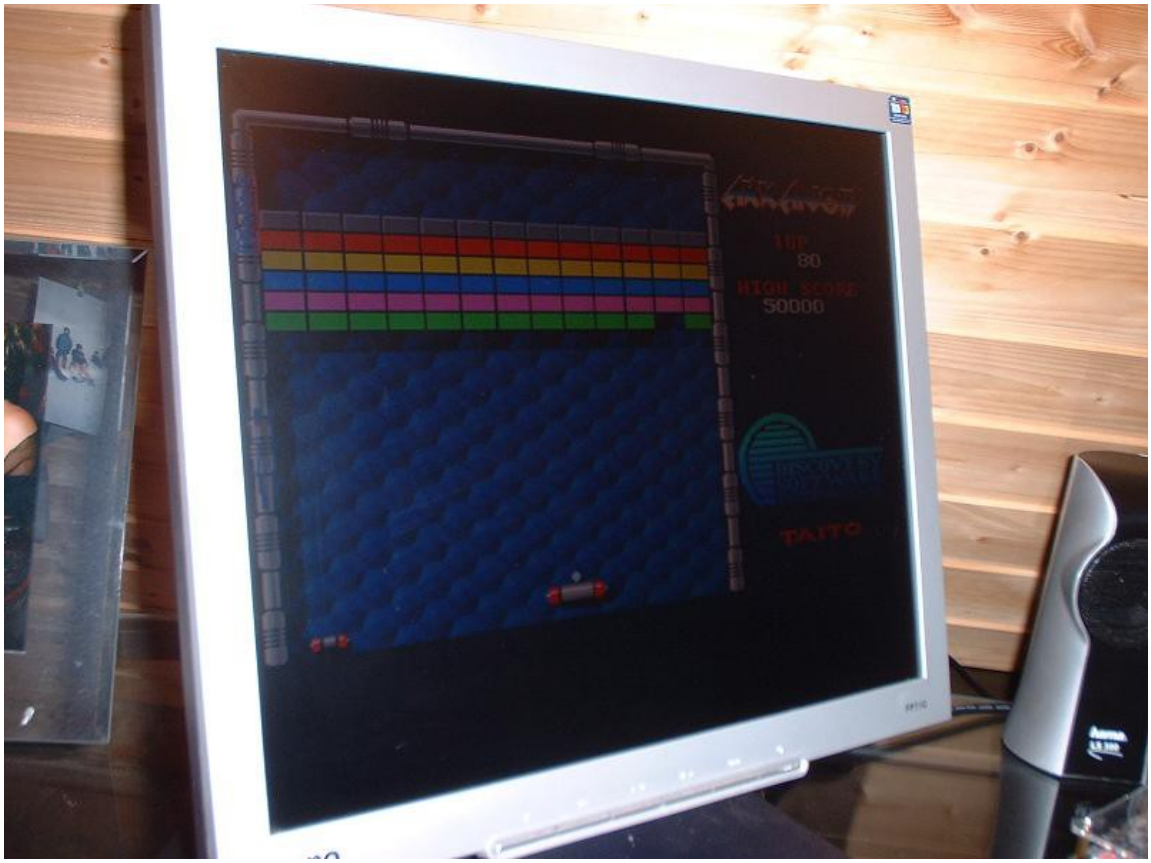
Sourcode

```
'-----------------------------------------------------------------------
'                           micro_kbd.BAS
'                  (c) 2008 peter.sieg1@gmx.de
'                  PS2 AT Keyboard emulator for Minimig
' NOTE THAT THE LIBRARY IS AN OPTIONAL ONE and by default is not included !
'-----------------------------------------------------------------------
$regfile = "m8def.dat"
$crystal = 1000000

$lib "mcsbyteint.lbx"                                       ' use optional
lib since we use only bytes

Config Portc = Input                                        ' port c for
taster
Portc = 255                                                 ' pullups on

'configure PS2 AT pins
Enable Interrupts                                           ' you need to
turn on interrupts yourself since an INT is used
Config Atemu = Int1 , Data = Pind.3 , Clock = Pinb.0
'                 ^------------------------ used interrupt
'                              ^------------ pin connected to DATA
'                                       ^-- pin connected to clock
'Note that the DATA must be connected to the used interrupt pin

Waitms 500                                                  ' optional
delay
Do
    Waitms 150
    If Pinc.0 = 0 Then Sendscankbd F12                ' send a scan code
    If Pinc.1 = 0 Then Sendscankbd Pgdn               ' send a scan code
    If Pinc.2 = 0 Then Sendscankbd Pgup               ' send a scan code
    If Pinc.3 = 0 Then Sendscankbd Pos1               ' send a scan code
    If Pinc.4 = 0 Then Sendscankbd F1                 ' send a scan code
    Waitms 150
Loop

F12:
Data 3 , &H07 , &HF0 , &H07

F1:
Data 3 , &H05 , &HF0 , &H05

Pgup:
Data 5 , &HE0 , &H7D , &HE0 , &HF0 , &H7D

Pgdn:
Data 5 , &HE0 , &H7A , &HE0 , &HF0 , &H7A

Pos1:
Data 5 , &HE0 , &H6C , &HE0 , &HF0 , &H6C
```

-> http://petersieg.webng.com/minimig/micro_kbd.zip



An einem Benq FP71G läuft Minimig/DE1 sehr gut



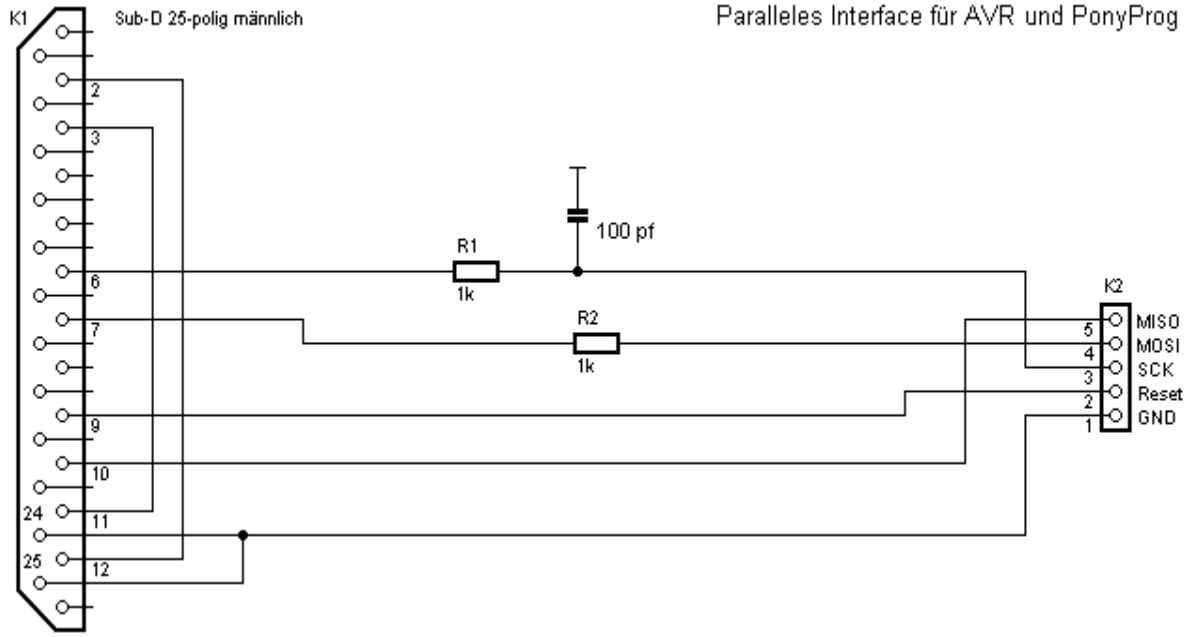
Links

http://home.hetnet.nl/~weeren001/
http://gamesource.groups.yahoo.com/group/minimigtg68/
http://www.amiga.org/modules/newbb/viewtopic.php?
topic_id=44173&forum=8&viewmode=flat&order=ASC&start=0
http://home.freeuk.com/fpgaarcade/index.htm

AVR Applikationen

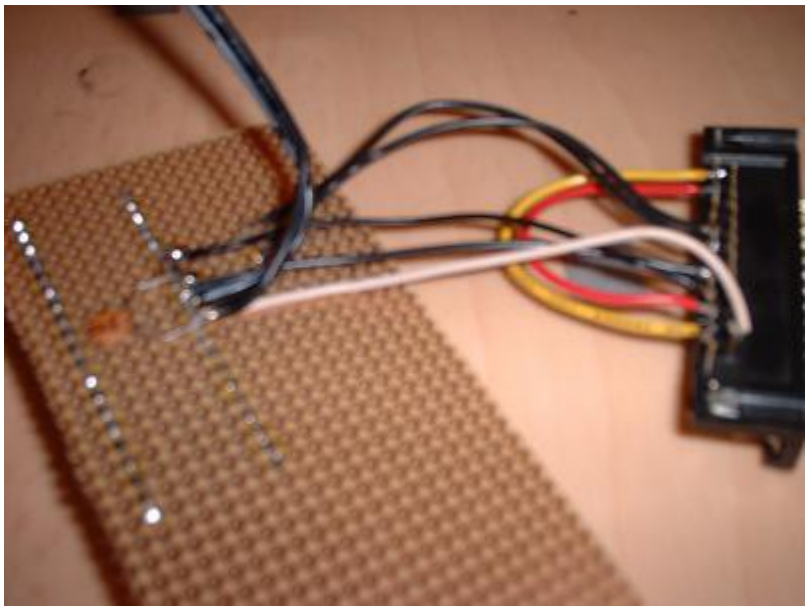
Einfachst Programmer

Ich wollte mal den einfachsten AVR Programmer nachbauen.. Das original Schaltbild ist auf der PonyProg Seite zu finden. Hatte ich nun gebaut, aber 'Write failed'..? Habe dann gegoogelt und gute Hinweise gefunden. Die Leitungen sollten möglichst kurz sein! Gegen Leitungsschwingen sollte ein Abblockkondensator in die SCK Leitung gegen GND rein. Printerport auf EPP/ECP, nicht SPP stellen. Habe ich gemacht und das ca. 80cm lange Parallelport-Verlängerungskabel dazwischen raus genommen.. und siehe da, es funktionierte! Wichtig ist, das sobald man auf externen Takt umstellt, man danach den AVR nicht mehr hiermit programmieren kann, da ja hier kein ext. Takt angeschlossen ist. Für mmc2iec ist das nicht wichtig, weil kein ext. Takt verwendet wird. Ansonsten kann man natürlich einen Quartz zw. XTAL1 + XTAL2 plus 2x22pf gegen GND für einen ext. Takt anschließen. Dieser einfachst Programmer soll STK200 kompatibel sein. Unter PonyProg ist Parallel - Avr ISP I/O und der richtige LPT Port einzustellen. Auch hier die Kalibrierung nicht vergessen! Geänderter Schaltplan:

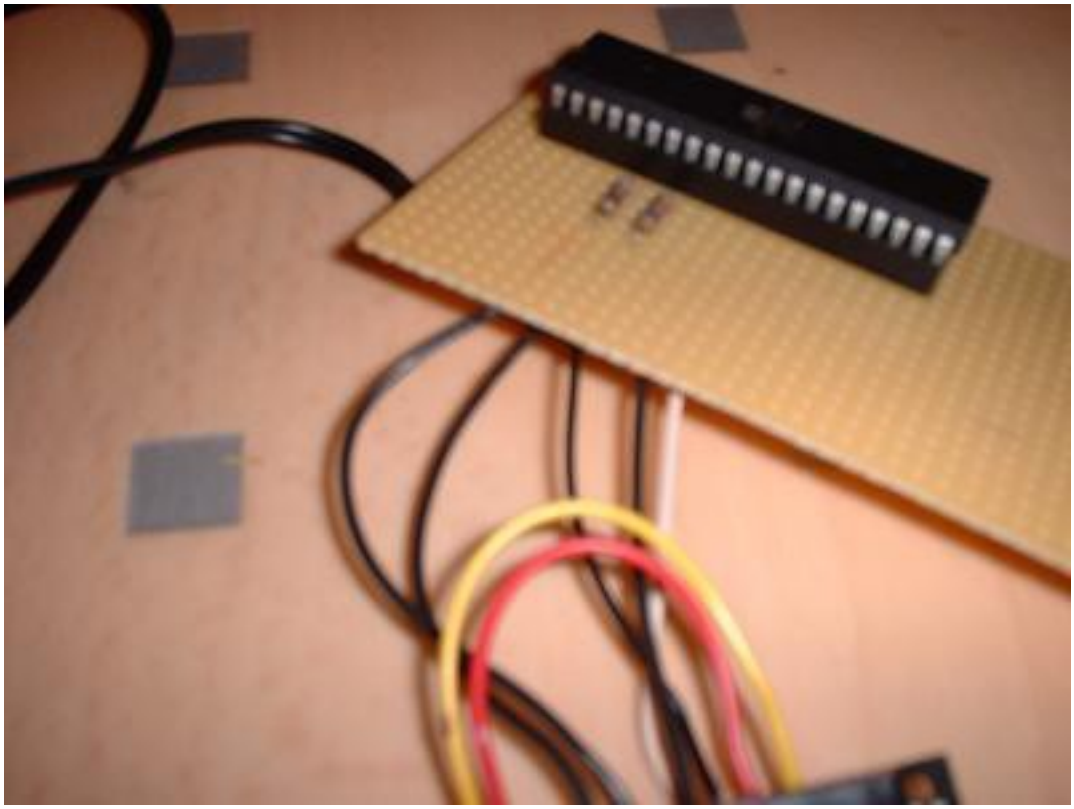


Lochrasteraufbau

Man sieht die beiden Kabelbrücken zwischen 2/12 und 3/11 direkt am Stecker.
Auf der Unterseite sieht man den Abblockkondensator (hier nur 22 pf).
Das helle Kabel zwischen Stecker und Platine ist GND:



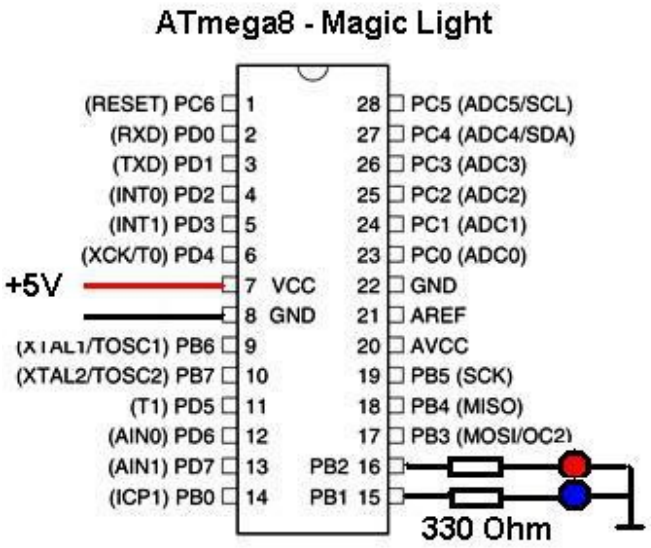
Von oben sind nur die beiden 1k Widerstände zu sehen:



Magic Light

Simple Hello World Schaltung mit ATmega8.
Einfache PWM - Bascom Anwendung. Ideal als erste Nachbauschaltung.
Läßt 2 LED abwechselt heller und dunkler werden (Fading mittels PWM).
Um gute 'Mischfarben' zu erhalten am besten eine Mehrfarb-LED nehmen.
Alternativ 2 einfache, möglichst helle LED's. LED's mit Schmirgelpapier anrauhen,
um die Lichtstreuung zu verstärken etc.

Schaltplan:



Link: http://petersieg.webng.com/avr/avr_pwm.zip
Quellcode:

```
$regfile = "m8def.dat"
$crystal = 8000000

Config Portb.1 = Output
Config Portb.2 = Output

Dim N As Byte

Config Timer1 = Pwm , Pwm = 8 , Compare A Pwm = Clear Up , Compare B Pwm =
Clear Up , Prescale = 1

Do
 For N = 0 To 255
  Compare1a = N
  Compare1b = 255 - N
  Waitms 150
 Next N
 Waitms 3000
 For N = 255 To 0 Step -1
  Compare1a = N
  Compare1b = 255 - N
  Waitms 150
 Next N
 Waitms 3000
Loop

End
```

Tiny Basic im ATmega8 auf Pollin Eval. Board

Basierend auf BSS02. Onne TV-Ausgabe. Kommunikation über serielle Schnittstelle mit 9600 8N1. Läuft so direkt auf einem Pollin Eval. Board mit ATmega8, 16MHz Quartz. Quellcode, HEX Datei, BSS02 orig. Seite sind im ZIP Archive.

Von der Firma Pollin (www.pollin.de) gibt es diese preiswerte AVR Entwicklungsboard für ca. 15€ als Bausatz. Zur Programmierung kann PonyProg verwendet werden.

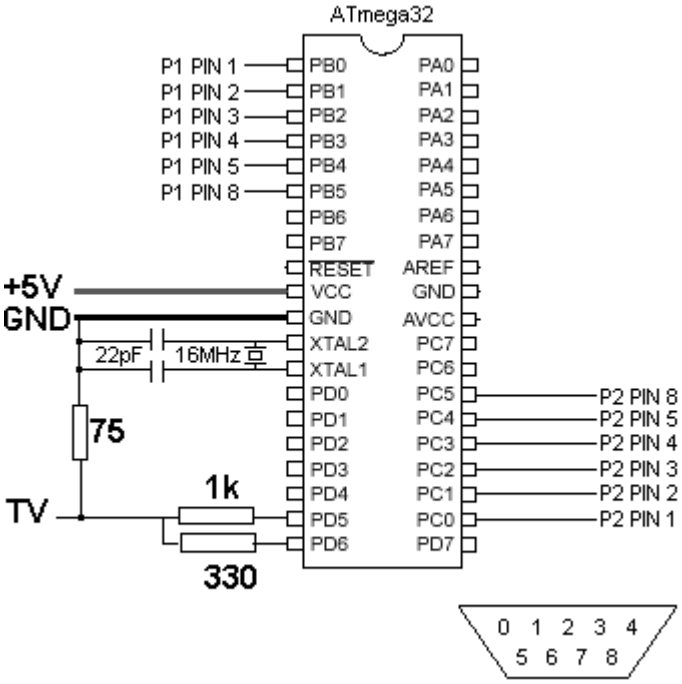
Das schöne an diesem Tiny Basic ist, das es auf einen Port von Li-Chen Wang's TinyBasic, aus dem Jahre 1976 (Veröffentlicht im Doctor Dobb's Journal) beruht! Von Jeff Hunsinger wurde es in die AVR Welt portiert. Ich habe nur ein paar kleinere Anpassungen für den Atmega8 gemacht. Außer 16MHz Quartzbeschaltung wird nur RX und TX benötigt.

Link: http://petersieg.webng.com/avr/avr_tinybasic.zip

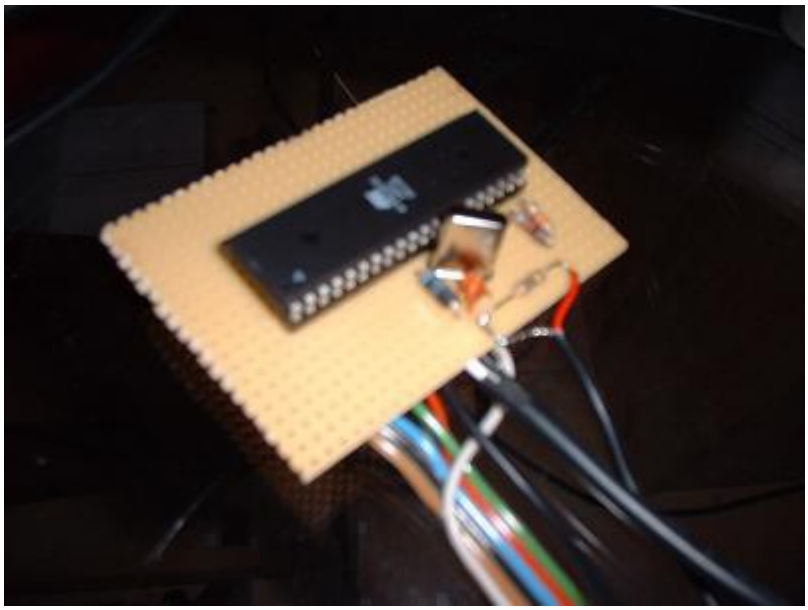
AVR Nova Strike (Galaxian)

Link: http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/s2004/kfm9/index.htm
Ein Space Invaders / Galaxian Style Spiel mit CodeVision C Sourcecode!
Zip-Archive inkl. HEX Datei (NTSC!): http://petersieg.webng.com/avr/NovaStrike.zip

Schaltplan:



Lochrasteraufbau:



Bildschirmfoto:

Fuses:

	Low	High	
Default:	0xE1	0x99	Apply default values
Current:	0xEF	0xD1	Apply user values

Low		High	
☐	BODLEVEL	☐	OCDEN
☐	BODEN	☐	JTAGEN
☐	SUT1	☑	SPIEN
☑	SUT0	☐	CKOPT
☐	CKSEL3	☑	EESAVE
☐	CKSEL2	☑	BOOTSZ1
☐	CKSEL1	☑	BOOTSZ0
☐	CKSEL0	☐	BOOTRST

AVR CP/M Minirechner

Links:
http://spritesmods.com/?art=avrcpm
http://www.mikrocontroller.net/articles/AVR_CP/M
http://www.mikrocontroller.net/topic/177481#1756046
http://avr.cwsurf.de/?AVR_CP%2FM
http://www.moria.de/~michael/cpmtools/

Jeroen Domburg (Sprite_tm) hat als ‚Proof-of-concept' aus einem Atmega88 AVR, einem gerade verügbarem 256x4 Dram und einer SD Karte eine minimalistischen CP/M Rechner geschaffen. Diese Projekt hatte mich sofort in seinen Bann gezogen und ich mußte es unbedingt nachbauen. Fast schon ungläubig schaute ich, als der Aufbau dann auch tatsächlich funktionierte! Im mikrocontroller Forum habe ich dann Gleichgesinnte gesucht, um das Projekt ggf. etwas weiter zu verbreiten und weiter zu entwicklen. Das Mini-CP/M System ist über serielle Konsole mit 38400 8N1 erreichbar (z.B Hyperterm/Minicom). Das diskimage wird ohne Filesystem einfach mit dd if=diskimage of=/dev/sdf1 bs=512 auf eine SD Karte gespielt (/dev/sdf1 ggf. durch anderen Bezeichner ersetzen).

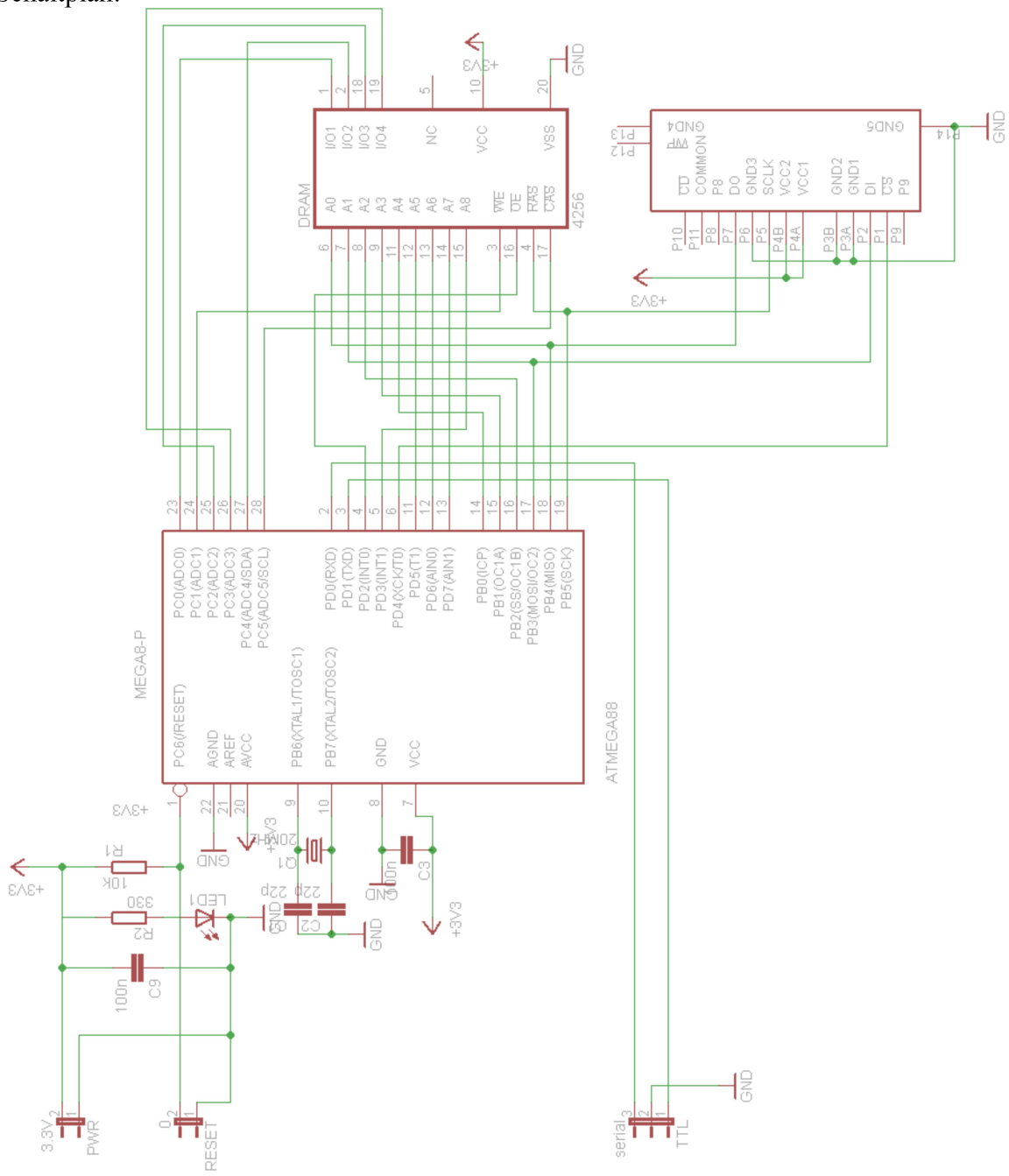
Mit den CPMTOOLS von Michael Haardt läßt sich das diskimage ändern:
Die Win32 Programme müssen zwingend im Wurzelverzeichnis entpackt werden, sodass \cpmtools entsteht.

Aufrufbeispiele:
cpmls -f avrcpm diskimage - zeigt Inhaltsverzeichnis an
cpmcp -f avrcpm diskimage datei.com 0:datei.com - kopiert datei.com in das image
cpmcp -f avrcpm diskimage 0:datei.com datei.com - kopiert datei.com aus dem image
cpmrm -f avrcpm datei.com - loescht datei.com aus dem image

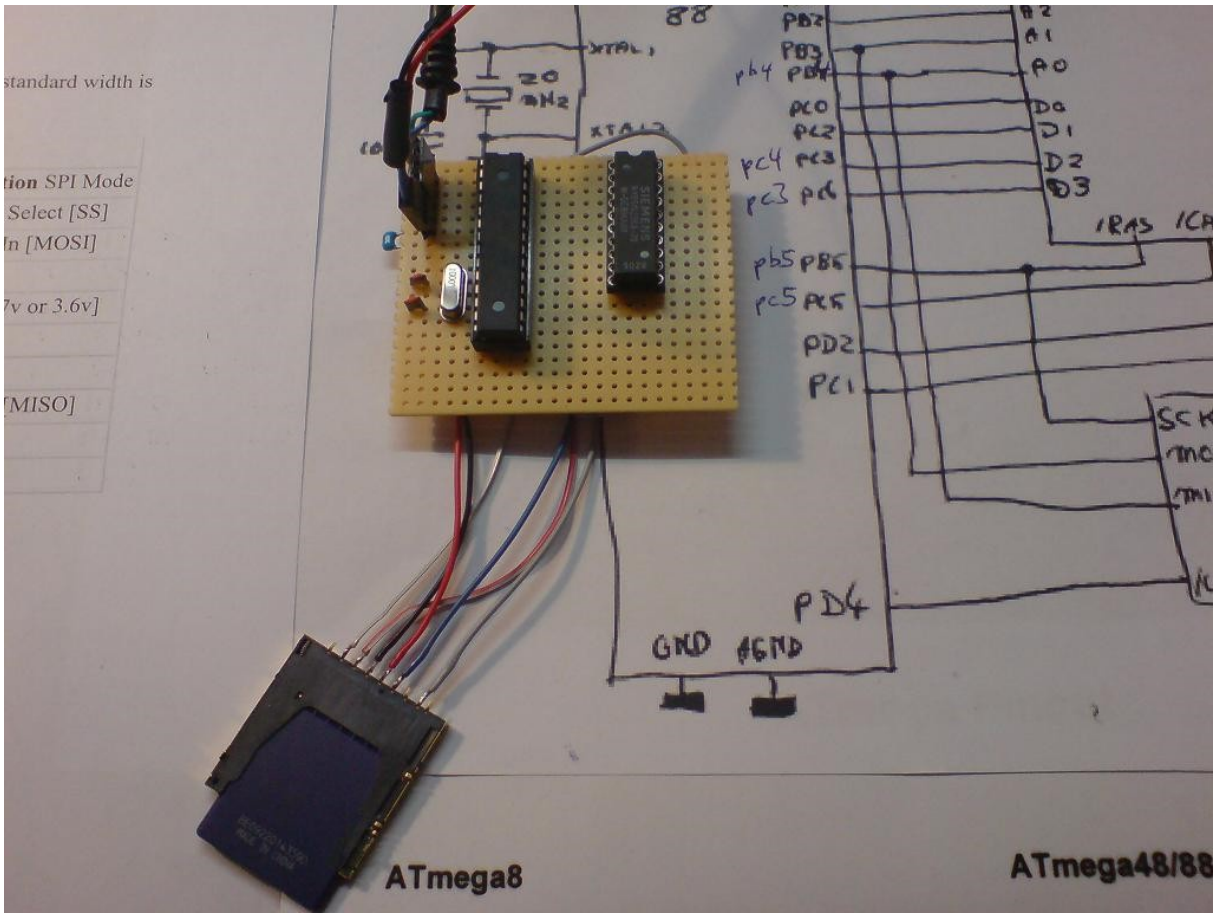
Verbindungsliste:

.equ ram_oe = 2 .equ ram_a8 = 3 .equ mmc_cs = 4 .equ ram_a5 = 5 .equ ram_a6 = 6 .equ ram_a7 = 7	
;Port B .equ ram_a4 = 0 .equ ram_a3 = 1 .equ ram_a2 = 2 .equ ram_a1 = 3 .equ mmc_mosi=3 .equ ram_a0 = 4 .equ mmc_miso=4 .equ ram_ras= 5 .equ mmc_sck= 5	;Port C .equ ram_d1 = 0 .equ ram_w = 1 .equ ram_d2 = 2 .equ ram_d4 =3 .equ ram_d3 = 4 .equ ram_cas=5

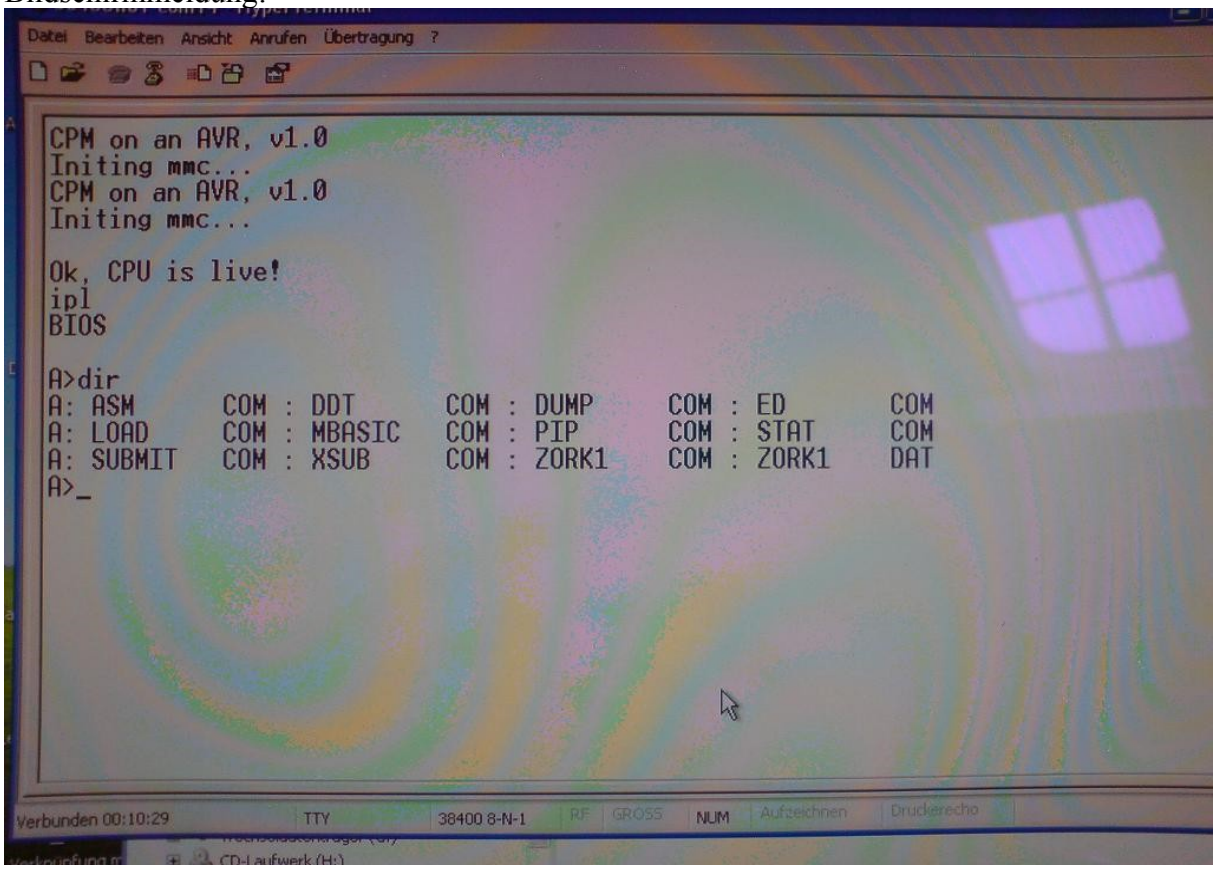
Schaltplan:


DRAM
4256
MEGA8-P
ATMEGA88
+3V3
GND
R1
10K
R2
330
LED1
C9
100n
C2
22p
C3
100n
Q1
20MHz
PWR
RESET
TTL
serial

Lochrasteraufbau:

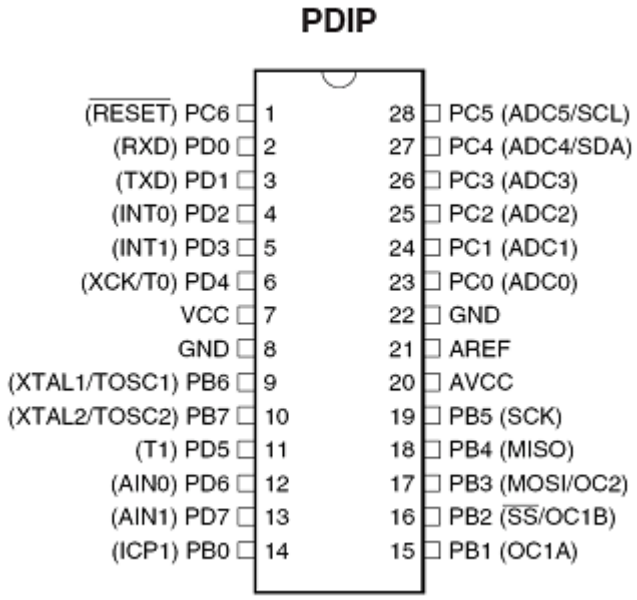


Bildschirmmeldung:



Bauteile Pinouts:

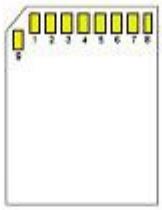
Atmega88:



Dram 256x4:

```
44256, 44258
256kx4 DRAM.

     +----------+
  D0 |1  +--+ 20| GND
  D1 |2        19| D3
 /WE |3        18| D2
/RAS |4        17| /CAS
     |5 44256 16| /OE
  A0 |6 44258 15| A8
  A1 |7        14| A7
  A2 |8        13| A6
  A3 |9        12| A5
 VCC |10       11| A4
     +----------+
```

SD Karte:



SD Card

The Secure Digital card dimensions are: 24mm wide x 32mm long. The standard width is 2.1mm, while the Thin SD Memory Card has a width 1.4mm.

Secure Digital Pinout			
Pin #	**Pin Name**	**SD Signal Function** SD Mode	**SPI Signal Function** SPI Mode
1	DAT3/CS	Data Line 3	Chip Select/Slave Select [SS]
2	CMD/DI	Command Line	Master Out/Slave In [MOSI]
3	VSS1	Ground	Ground
4	Vdd	Voltage Supply [2.7v or 3.6v]	Voltage Supply [2.7v or 3.6v]
5	Clock	Clock	Clock [SCK]
6	Vss2	Ground	Ground
7	DAT0/D0	Data Line 0	Master In Slave Out [MISO]
8	DAT1/IRQ	Data Line 1	Unused or IRQ
9	DAT2/NC	Data Line 2	Unused

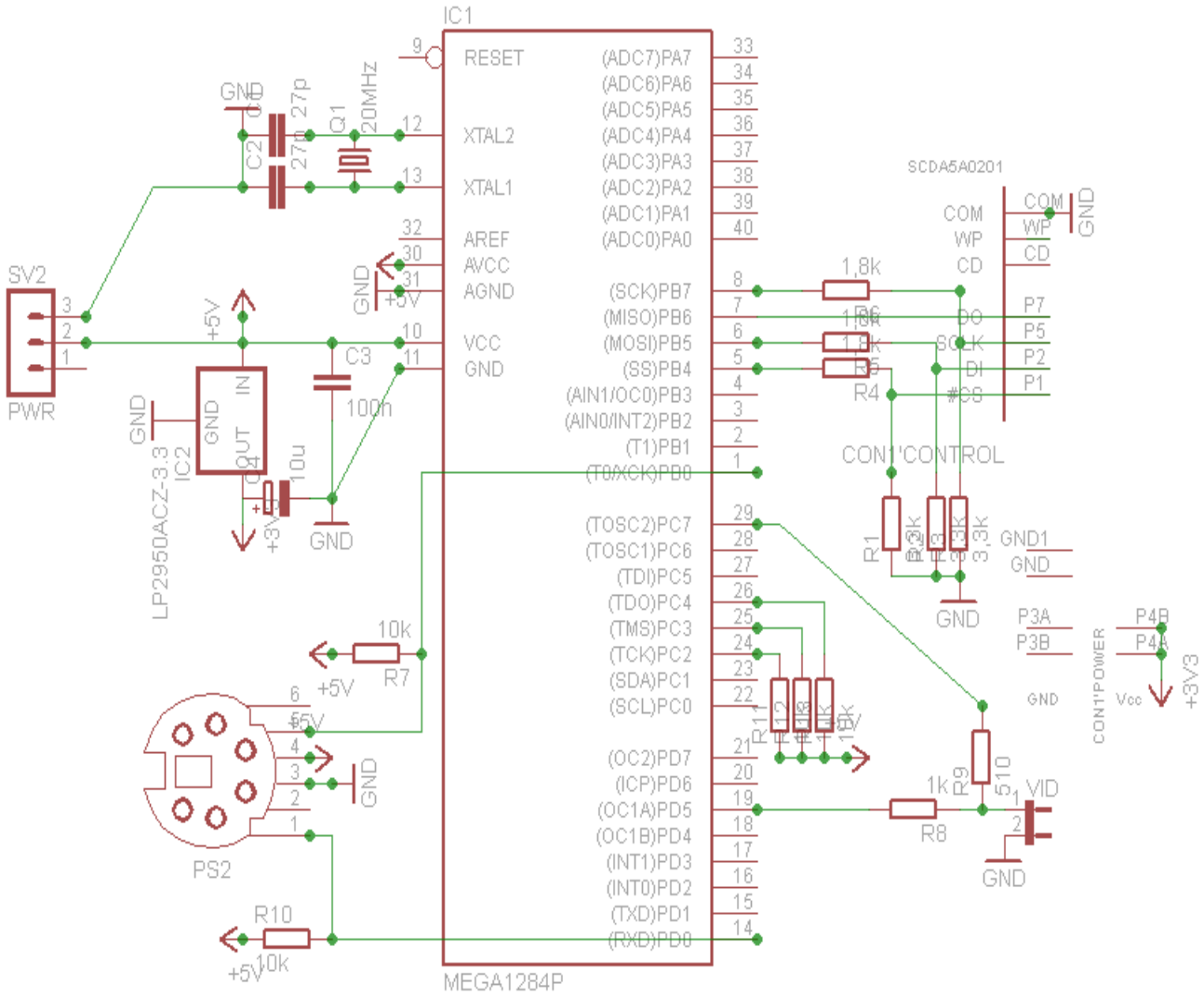
AX81 – ZX81 im AVR

Von Jörg Wolfram

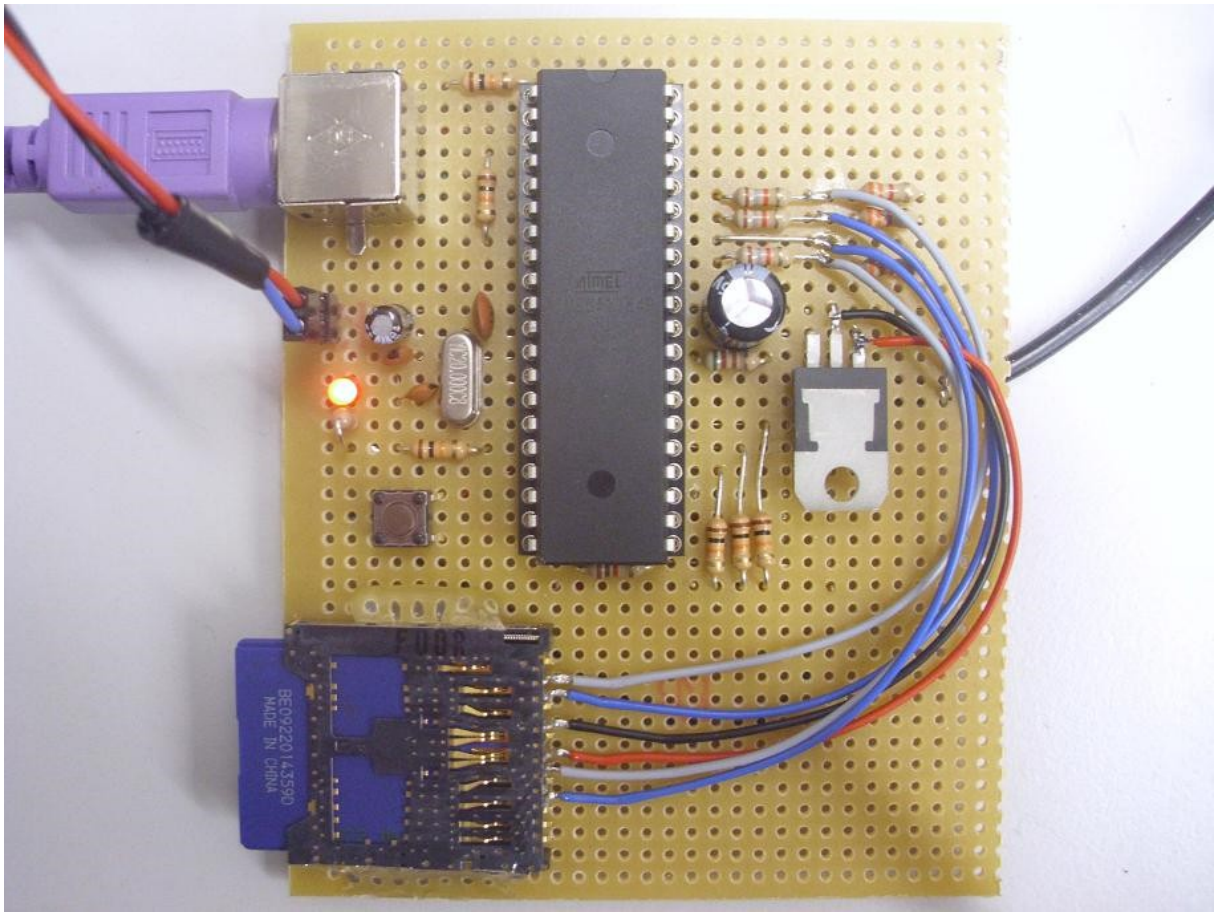
Links:
http://www.jcwolfram.de/projekte/avr/ax81/main.php
http://www.mikrocontroller.net/topic/236397

Jörg Wolfram hat es 2011 geschafft, eine komplette Z80 und ZX81 Emulation in einem ATmega644/1284P unterzubringen. Anzeige über Videoout, VGA oder sogar LCD Anbindung. Eingabe über PS/2 Tastatur und als Massenspeicher kann eine SD Karte dienen auf dem ein spezielles Tapeimage von 132MB über 8000 16k Programme speichern kann.

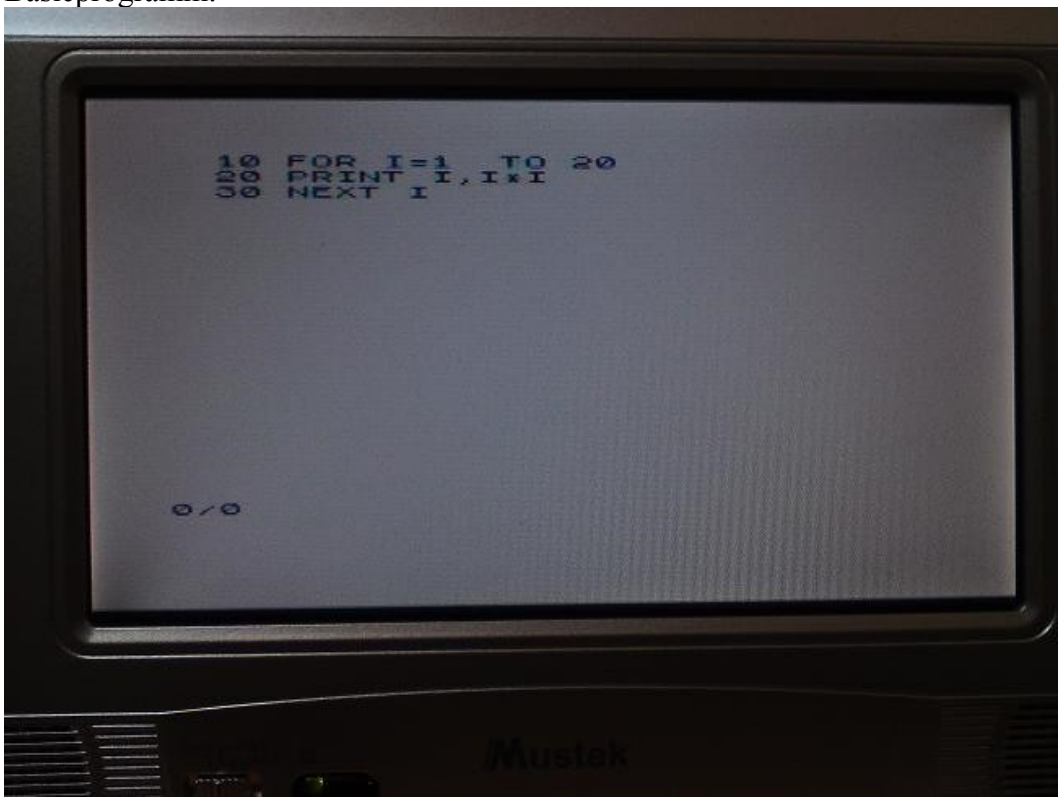
Schaltplan:



Lochrasteraufbau:



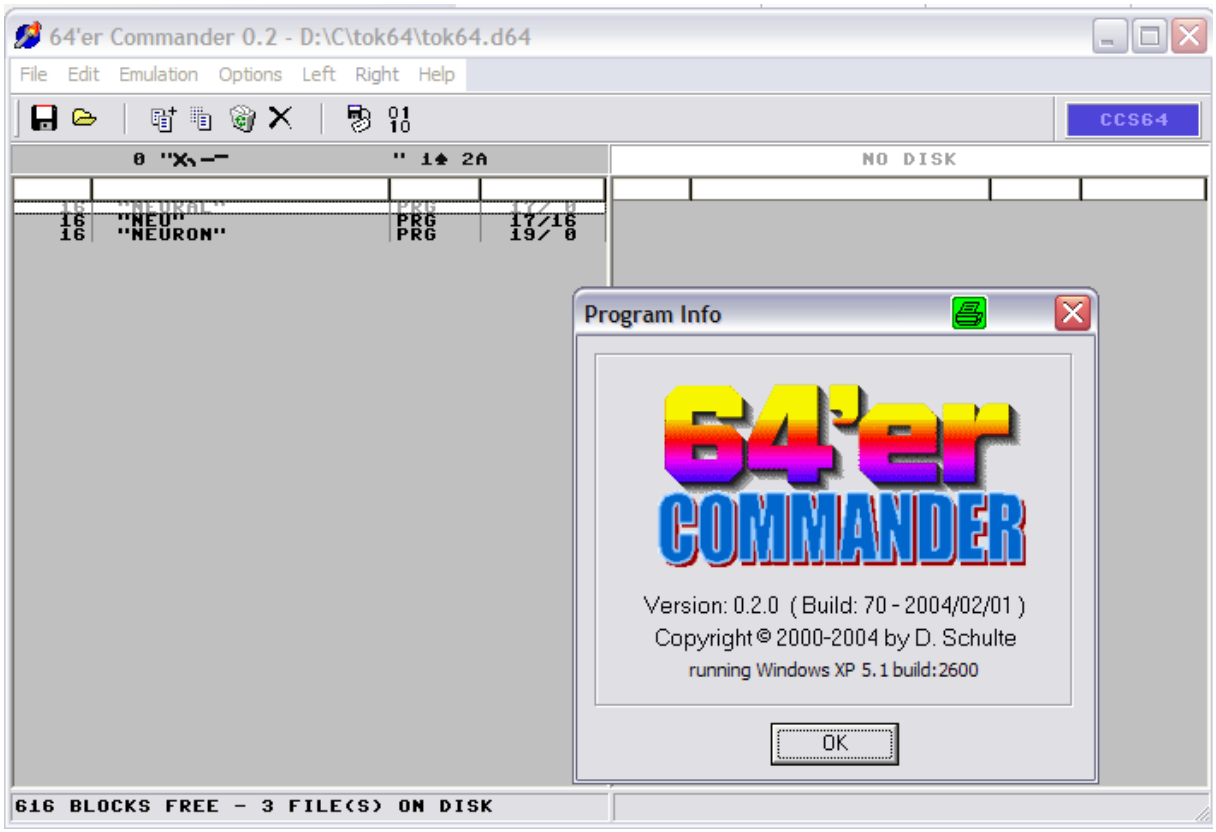
Basicprogramm:



64'er Commander

Link: http://c64.revivalteam.de

Mit dem Programm lassen sich z.B D64 Diskimages öffnen, erzeugen, Dateien zwischen Images kopieren, verschieben, löschen etc. Außerdem lassen sich einzelnen Programme aus Diskimages extrahieren.



64Copy

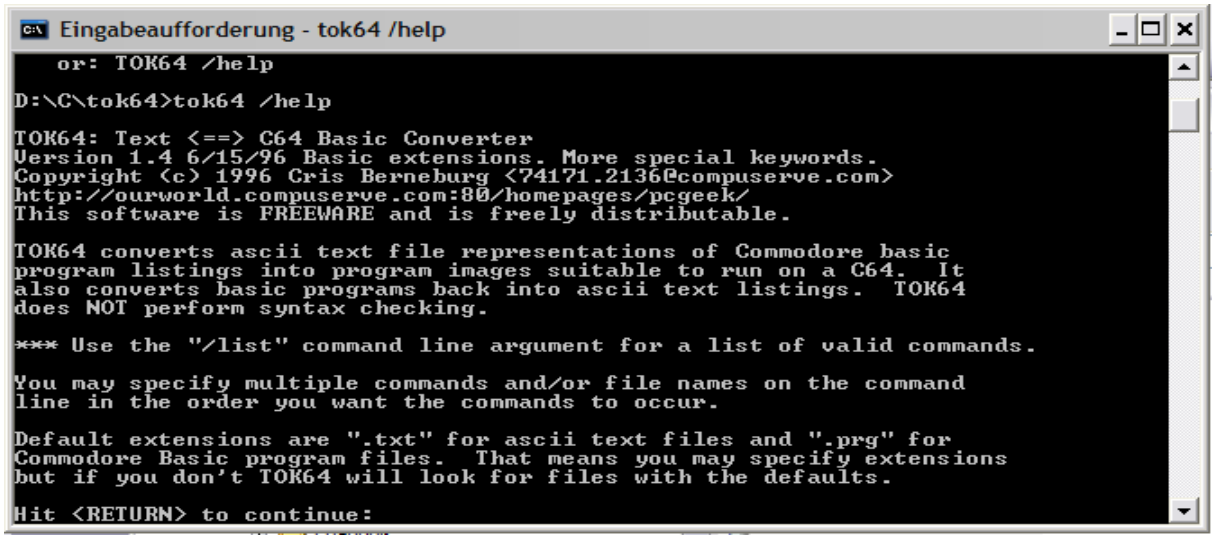
Copyright by Peter Schepers 1994-2005

Hiermit lassen sich viele Formatumwandlungen durchführen. Also zum Beispiel: PRG in T64
Es enthält noch viele weitere Funktionen und ist das Allzwecktool für Formatkonvertierungen.

Tok64 – Konvertierung ASCII-Text <-> Basic Token

Copyright (c) 1996 Cris Berneburg <74171.2136@compuserve.com>
http://ourworld.compuserve.com:80/homepages/pcgeek/
This software is FREEWARE and is freely distributable.

Zum Wandeln von Basic Programmen in ASCII Text und umgekehrt.



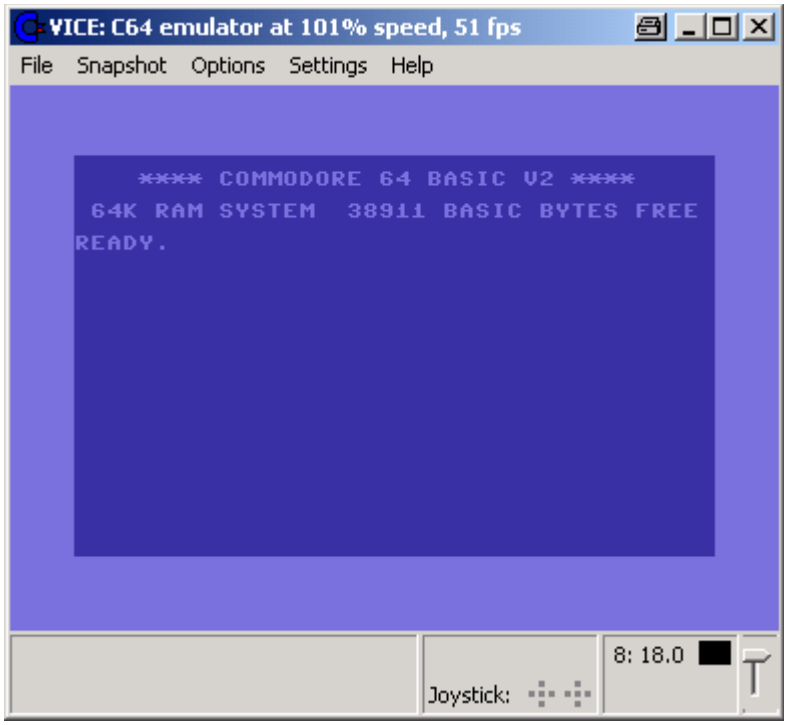
Emulationen

Wer einfach mal ein bißchen Retro-Feeling auf seinem PC haben möchte, für den sind sicher auch entsprechende Emulatoren interessant.

VICE

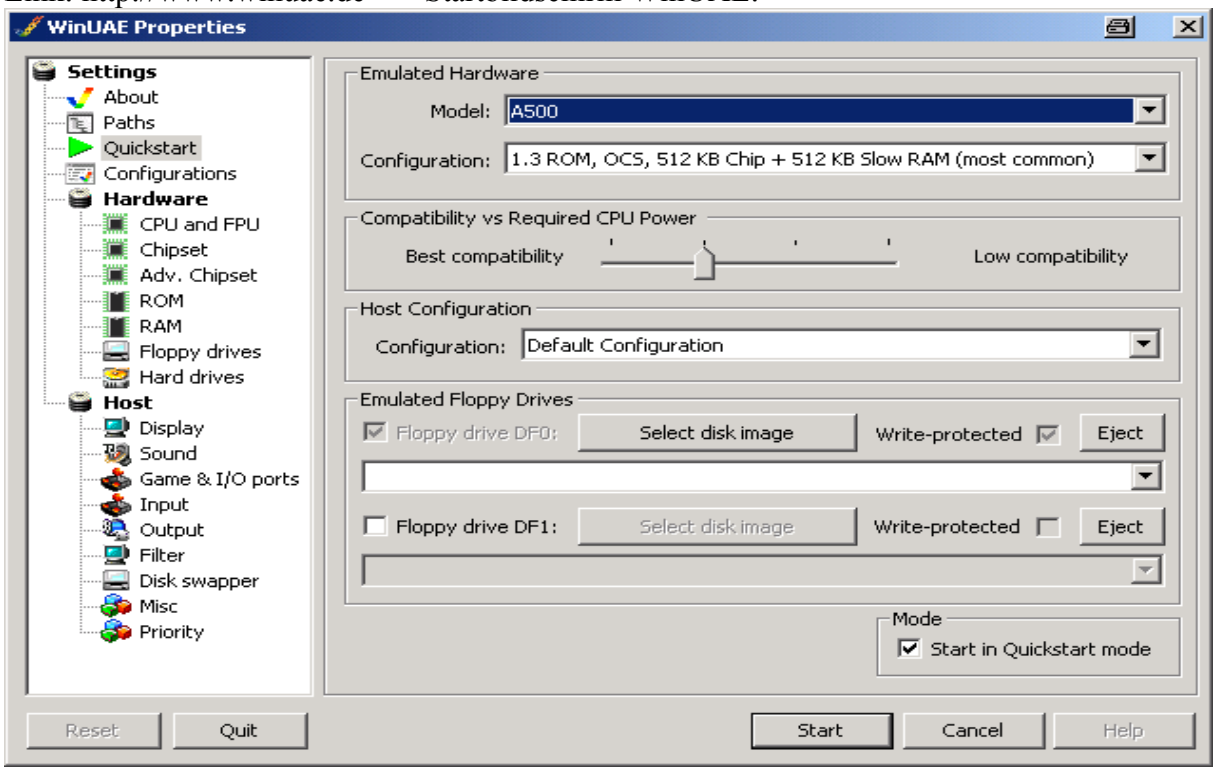
Der VICE Commodore 8-bit Emulator ist für mich einer der besten Emulatoren für PET/CBM, VC20, C64 und die 264er Reihe (C16,+4) von Commodore. Open Source und für viele Plattformen ist ein sog. Port verfügbar.

Link: http://www.viceteam.org/

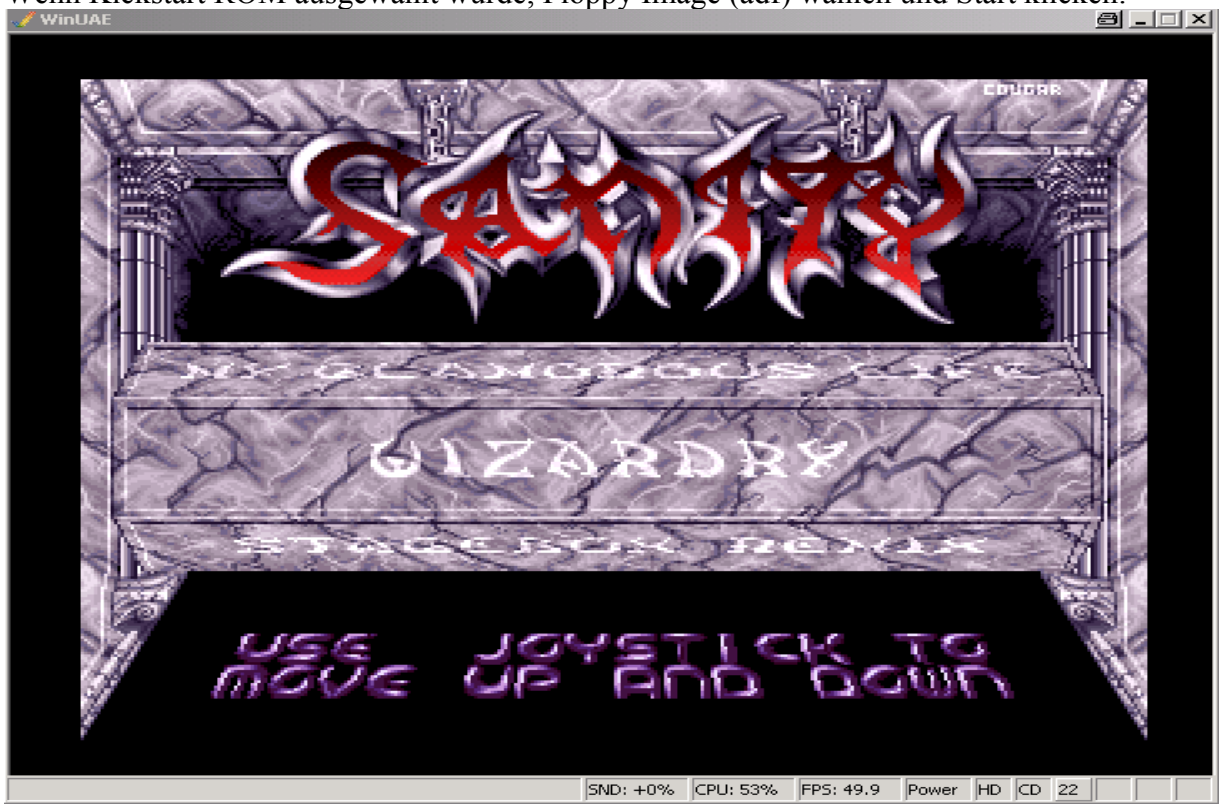


(Win)UAE

Universal **A**miga **E**mulator
Link: http://www.winuae.de - Startbildschirm WinUAE:



Wenn Kickstart ROM ausgewählt wurde, Floppy Image (adf) wählen und Start klicken:



Weiterführende Adressen

Verein zum Erhalt klassischer Computer e.V

http://www.classic-computing.de

Der Verein veranstaltet einmal im Jahr die sog. Classic Computing. Auf dieser Austellung können eine Vielzahl von teilweise sehr seltenen Exponaten bewundert werden. Eine Vielzahl der Systeme wird im funktionsfähigem Zustand gezeigt. Auch kann Mann/Frau einmal selbst Hand anlegen und einen der alten Klassiker an original Hardware ‚zocken'.

Gruppenbild von der Classic Computing 2007:

Forum64

http://www.forum64.de

DIE Anlaufstelle für Diskussionen, Hilfestellungen etc. um C64, Amiga und andere(s).

C64 Wiki

http://www.c64-wiki.de – Hier wächst Wissen – auch mit ihrer Unterstützung!

A1k.org

http://www.a1k.org

DIE Anlaufstelle für alle Amiga Fan’s.

Kleine Bauteilkunde

Widerstände

Farbe	Ziffer 1-3	Multiplikator	Toleranz
Silber		0.01	10%
Gold		0.1	5%
Schwarz	0	1	20%
Braun	1	10	1%
Rot	2	100	2%
Orange	3	1k	
Gelb	4	10k	
Grün	5	100k	
Blau	6	1M	
Violett	7	10M	
Grau	8	100M	
weiß	9	1G	

Beispiele:
Gelb – Violett – Rot – Gold = 4,7k Ohm 5%
Braun – Schwarz – Orange – Gold = 10k Ohm 5%

SMD Widerstände

Sind mit 3 Ziffern beschriftet. Die ersten beiden Ziffern geben den Wert an, die dritte Ziffer den Multiplikator. Bei 4 Ziffern geben die ersten drei den Wert an und die vierte Ziffer den Multiplikator.

Beispiel:
$103 = 10 * 10^3 = 10k$ Ohm
$220 = 22 * 1 = 220$ Ohm

ELKO’S

Elektrolytkondensatoren sind gepolte Bauteile. Daher ist hier, wie bei allen gepolten Bauteilen auf den korrekten Einbau zu achten!

Der Minuspol ist bei diesen Bauteilen gekennzeichnet:

Tantal-Elko’s - Hier ist der Pluspol mit einem ‚+’ markiert.

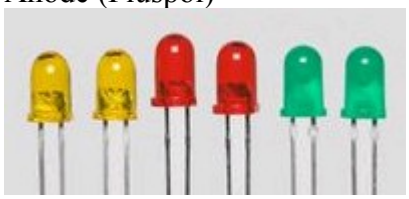
SMD-Elko’s – Hier ist der Pluspol abgeflacht.

Dioden.

Bei diesen gepolten Bauteilen ist der Minuspol (Kathode) durch einen Strich markiert.

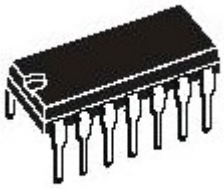
Leuchtdioden

Hier ist der Minuspol (Kathode) am Gehäuse abgeflacht. Das längere Anschlußbein ist die Anode (Pluspol)



Integrierte Schaltungen

Hier ist auf die Einbaurichtung der Kerbe zu achten. Fassungen sollten ebenfalls mit einer korrekten Position der Kerbe verbaut werden, sodaß es später nicht zu einer falschen Bestückung kommt!



Quarzoszillator - Im DIP14 Gehäuse.

Pin 1 eines Quarzoszillators ist mit einem Punkt an der Gehäuseoberseite oder mit einer anders geformten Ecke gekennzeichnet. Elektrisch hat der Pin1 keine Bedeutung. Der Pin rechts daneben ist bei DIL 14 Pin 7 und bei DIL 8 Pin 4 (Zählung wie beim Standard-DIL-Gehäuse). Dieses ist der GND (Masse) Pin. Der Pin über GND ist bei DIL 14 Pin 8 und bei DIL 8 Pin 5. An ihm liegt das Takt-Ausgangssignal an. Der Pin oberhalb der markierten Ecke (links – oben) ist bei DIL 14 Pin 14 und bei DIL 8 Pin 8. Er ist die Versorgungsspannung (meist +5 V).

TIP:
Noch ein genereller Tip, falls auf einer Platine mal die Polung von Bauteilen nicht klar ersichtlich sein sollte. Die Pins gegen einen bekannten Masse (GND) Punkt auf Durchgang messen. Das ist dann der Minus/Masse/GND Anschluß.

Danksagungen

Jogi (Joachim Nemetz) und Jens Schönfeld für die C64 Ersatz-PLA auf Eprom 27512 Schaltung und Angaben.
JerryMaus (Andreas Senk) für seine C64 –Assy 250407 Restaurationsstory.
Pentagon (Thomas Wirtzmann) für seine C64 Restaurationsstory.
Area51HT (Peter Trachte) für seinen Großumbau, Schutzschaltung und weitere Tips.
X1541 (Nicolas Welte) und enthusi (Martin Wendt) für NeoRAM + Treiber.
Ing. Thomas Winkler (Diddl) für die Entwicklung des XS1541 Adapters und der Final Expansion für den VC20.
Thomas Giesel (skoe) für die Entwicklung von Easyflash für den C64
Björn 'JMP$FCE2' Wieck für die S-Video Schaltung am C64, die Platinenroutings zu anderen Projekten (z.B EasyFlash) und für Tiny Eprommer nachbau.
Slawomir Musial (Swinkels) und Ingmar Weigel für die Entwicklung und Erweiterung von SwinSIDX2 und Micro SwinSID88.
Klaus Hegemann für die Pinout Foto's und Angaben im DTV Kapitel.
Gerrit Heitsch für die C64 Oszilloskop Messpunkte.
Daniel Svegert / MadModder für sein Übersichtsbild der DTV Anschlußpunkte.
Bencao74 (Jochen Zurborg) für sein DTV Mod Foto.
Roland Tögel für Easyflash – Einfaches Flash-Tool für das DTV.
Dennis van Weeren für seine tolle Leistung, einen Amiga 500 in FPGA zu realisieren.
Tobias Gubener (tobiflex) für seinen 68000 Core und Minimig auf DE1/2 Portierung.
Jan Derogee für die 1541-III – SD Karte am C64 auf PIC Basis.
Lars Pontoppidan für seine MMC2IEC Umsetzung der 1541-III auf AVR Atmega32 Basis.
Ingo Korb für die Weiterentwicklung der Firmware zum SD2IEC.
Gideon Zweijtzer für die Entwicklung der 1541 Ultimate. Jan Janssen (Cyberdyne) für Bild und Text zur 1541U.
Tom Kirk für sein PS/2 Mausadapter für Amiga & Atari ST.
Martin Clausen für die freundliche Unterstützung zum PS/2 Tastaturadapter für Amiga.
Sascha van Wahnem (AmigaGTI) für die Angaben zum ELKO Tausch an SMD Amiga's.
Stefan Pitsch, ^TePe^ und Stefan Hohendorf (HONI) für die PC Floppy Umbauten für Amiga.
Jopower und The_Killer für den A500+ auf 2MB Chipram Umbau.
Jean-Francois del Nero für die Entwicklung der HxC Floppy Emulatoren.
Richard Stallmann für ‚seine' GNU General Public License und die Open Source Bewegung.
Der Wikipedia und seinen unzähligen Authoren und Helfern.

Und last but not least, meiner Familie: Heike, Robin und Janis, für die Gedult und die Stunden, die ich mit alten Computern verbracht habe, anstatt mit ihnen. Und danken möchte ich an dieser Stelle auch meinen Eltern Edith und Reinold.

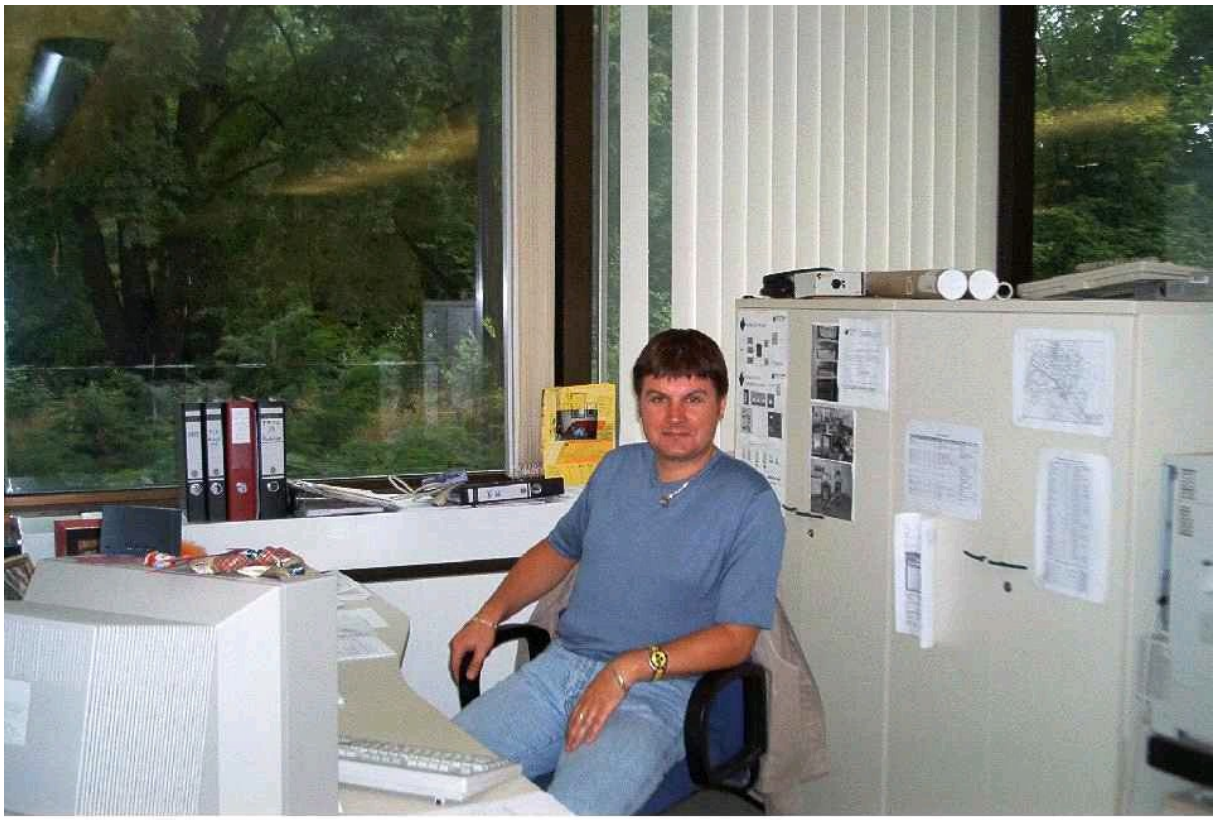
Über den Author

Mein Name ist Peter Sieg. Ich bin am 26.12.1963 in Boffzen geboren. Im Jahre 2008 bin ich somit 44 Jahre jung. An Homecomputern hatte ich früher mal ZX81 und Atari ST. Mit C64 und Amiga hatte ich zu damaligen Zeiten wenige Berührungen.

Nach der Ausbildung zum Chemikanten, habe ich Dosiersysteme betreut. Schulungen in Assembler und Fortran auf DEC PDP-11 folgten ABAP unter SAP R/2, später R/3. Grundkentnisse in Pascal, C, VB und Java sind inzwischen dazugekommen.

Generell geht es mir allerdings gegen den Strich, noch brauchbare Sachen einfach wegzuwerfen. Dabei hat es mir insbesondere alte Computertechnik angetan.

Diese ‚alte' Technik ist hoch interessant und durch die Spieleorientierung sind auch die eigenen Kinder nicht ganz außen vor.



Ein Bild von mir an meinem Arbeitsplatz...

Web Seiten:
http://petersieg.bplaced.net